

Apostolos Antonacopoulos ·  
Subhasis Chaudhuri · Rama Chellappa ·  
Cheng-Lin Liu · Saumik Bhattacharya ·  
Umapada Pal (Eds.)

LNCS 15315

# Pattern Recognition

27th International Conference, ICPR 2024  
Kolkata, India, December 1–5, 2024  
Proceedings, Part XV

15 Part XV

ICPR  
2024 INDIA



 Springer

MOREMEDIA 

# Lecture Notes in Computer Science

15315

## Founding Editors

Gerhard Goos  
Juris Hartmanis

## Editorial Board Members

Elisa Bertino, *Purdue University, West Lafayette, IN, USA*

Wen Gao, *Peking University, Beijing, China*

Bernhard Steffen , *TU Dortmund University, Dortmund, Germany*

Moti Yung , *Columbia University, New York, NY, USA*

The series Lecture Notes in Computer Science (LNCS), including its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), has established itself as a medium for the publication of new developments in computer science and information technology research, teaching, and education.


LNCS enjoys close cooperation with the computer science R & D community, the series counts many renowned academics among its volume editors and paper authors, and collaborates with prestigious societies. Its mission is to serve this international community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings. LNCS commenced publication in 1973.


Apostolos Antonacopoulos ·  
Subhasis Chaudhuri · Rama Chellappa ·  
Cheng-Lin Liu · Saumik Bhattacharya ·  
Umapada Pal  
Editors


# Pattern Recognition

27th International Conference, ICPR 2024  
Kolkata, India, December 1–5, 2024  
Proceedings, Part XV

*Editors*


Apostolos Antonacopoulos   
University of Salford  
Salford, Lancashire, UK

Rama Chellappa   
Johns Hopkins University  
Baltimore, MD, USA

Saumik Bhattacharya   
IIT Kharagpur  
Kharagpur, West Bengal, India

Subhasis Chaudhuri   
Indian Institute of Technology Bombay  
Mumbai, Maharashtra, India

Cheng-Lin Liu   
Chinese Academy of Sciences  
Beijing, China

Umapada Pal   
Indian Statistical Institute Kolkata  
Kolkata, West Bengal, India

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78353-1

ISBN 978-3-031-78354-8 (eBook)

<https://doi.org/10.1007/978-3-031-78354-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license  
to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

## President's Address

On behalf of the Executive Committee of the International Association for Pattern Recognition (IAPR), I am pleased to welcome you to the 27th International Conference on Pattern Recognition (ICPR 2024), the main scientific event of the IAPR.

After a completely digital ICPR in the middle of the COVID pandemic and the first hybrid version in 2022, we can now enjoy a fully back-to-normal ICPR this year. I look forward to hearing inspirational talks and keynotes, catching up with colleagues during the breaks and making new contacts in an informal way. At the same time, the conference landscape has changed. Hybrid meetings have made their entrance and will continue. It is exciting to experience how this will influence the conference. Planning for a major event like ICPR must take place over a period of several years. This means many decisions had to be made under a cloud of uncertainty, adding to the already large effort needed to produce a successful conference. It is with enormous gratitude, then, that we must thank the team of organizers for their hard work, flexibility, and creativity in organizing this ICPR. ICPR always provides a wonderful opportunity for the community to gather together. I can think of no better location than Kolkata to renew the bonds of our international research community.

Each ICPR is a bit different owing to the vision of its organizing committee. For 2024, the conference has six different tracks reflecting major themes in pattern recognition: Artificial Intelligence, Pattern Recognition and Machine Learning; Computer and Robot Vision; Image, Speech, Signal and Video Processing; Biometrics and Human Computer Interaction; Document Analysis and Recognition; and Biomedical Imaging and Bioinformatics. This reflects the richness of our field. ICPR 2024 also features two dozen workshops, seven tutorials, and 15 competitions; there is something for everyone. Many thanks to those who are leading these activities, which together add significant value to attending ICPR, whether in person or virtually. Because it is important for ICPR to be as accessible as possible to colleagues from all around the world, we are pleased that the IAPR, working with the ICPR organizers, is continuing our practice of awarding travel stipends to a number of early-career authors who demonstrate financial need. Last but not least, we are thankful to the Springer LNCS team for their effort to publish these proceedings.

Among the presentations from distinguished keynote speakers, we are looking forward to the three IAPR Prize Lectures at ICPR 2024. This year we honor the achievements of Tin Kam Ho (IBM Research) with the IAPR's most prestigious King-Sun Fu Prize "for pioneering contributions to multi-classifier systems, random decision forests, and data complexity analysis". The King-Sun Fu Prize is given in recognition of an outstanding technical contribution to the field of pattern recognition. It honors the memory of Professor King-Sun Fu who was instrumental in the founding of IAPR, served as its first president, and is widely recognized for his extensive contributions to the field of pattern recognition.

The Maria Petrou Prize is given to a living female scientist/engineer who has made substantial contributions to the field of Pattern Recognition and whose past contributions, current research activity and future potential may be regarded as a model to both aspiring and established researchers. It honours the memory of Professor Maria Petrou as a scientist of the first rank, and particularly her role as a pioneer for women researchers. This year, the Maria Petrou Prize is given to Guoying Zhao (University of Oulu), “for contributions to video analysis for facial micro-behavior recognition and remote bio-signal reading (RPPG) for heart rate analysis and face anti-spoofing”.

The J.K. Aggarwal Prize is given to a young scientist who has brought a substantial contribution to a field that is relevant to the IAPR community and whose research work has had a major impact on the field. Professor Aggarwal is widely recognized for his extensive contributions to the field of pattern recognition and for his participation in IAPR's activities. This year, the J.K. Aggarwal Prize goes to Xiaolong Wang (UC San Diego) “for groundbreaking contributions to advancing visual representation learning, utilizing self-supervised and attention-based models to establish fundamental frameworks for creating versatile, general-purpose pattern recognition systems”.

During the conference we will also recognize 21 new IAPR Fellows selected from a field of very strong candidates. In addition, a number of Best Scientific Paper and Best Student Paper awards will be presented, along with the Best Industry Related Paper Award and the Piero Zamperoni Best Student Paper Award. Congratulations to the recipients of these very well-deserved awards!

I would like to close by again thanking everyone involved in making ICPR 2024 a tremendous success; your hard work is deeply appreciated. These thanks extend to all who chaired the various aspects of the conference and the associated workshops, my ExCo colleagues, and the IAPR Standing and Technical Committees. Linda O’Gorman, the IAPR Secretariat, deserves special recognition for her experience, historical perspective, and attention to detail when it comes to supporting many of the IAPR’s most important activities. Her tasks became so numerous that she recently got support from Carolyn Buckley (layout, newsletter), Ugur Halici (ICPR matters), and Rosemary Stramka (secretariat). The IAPR website got a completely new design. Ed Sobczak has taken care of our web presence for so many years already. A big thank you to all of you!

This is, of course, the 27th ICPR conference. Knowing that ICPR is organized every two years, and that the first conference in the series (1973!) pre-dated the formal founding of the IAPR by a few years, it is also exciting to consider that we are celebrating over 50 years of ICPR and at the same time approaching the official IAPR 50th anniversary in 2028: you’ll get all information you need at ICPR 2024. In the meantime, I offer my thanks and my best wishes to all who are involved in supporting the IAPR throughout the world.

September 2024

Arjan Kuijper  
President of the IAPR

# Preface

It is our great pleasure to welcome you to the proceedings of the 27th International Conference on Pattern Recognition (ICPR 2024), held in Kolkata, India. The city, formerly known as ‘Calcutta’, is the home of the fabled Indian Statistical Institute (ISI), which has been at the forefront of statistical pattern recognition for almost a century. Concepts like the Mahalanobis distance, Bhattacharyya bound, Cramer–Rao bound, and Fisher–Rao metric were invented by pioneers associated with ISI. The first ICPR (called IJCPD then) was held in 1973, and the second in 1974. Subsequently, ICPR has been held every other year. The International Association for Pattern Recognition (IAPR) was founded in 1978 and became the sponsor of the ICPR series. Over the past 50 years, ICPR has attracted huge numbers of scientists, engineers and students from all over the world and contributed to advancing research, development and applications in pattern recognition technology.

ICPR 2024 was held at the Biswa Bangla Convention Centre, one of the largest such facilities in South Asia, situated just 7 kilometers from Kolkata Airport (CCU). According to ChatGPT “Kolkata is often called the ‘Cultural Capital of India’. The city has a deep connection to literature, music, theater, and art. It was home to Nobel laureate Rabindranath Tagore, and the Bengali film industry has produced globally renowned filmmakers like Satyajit Ray. The city boasts remarkable colonial architecture, with landmarks like Victoria Memorial, Howrah Bridge, and the Indian Museum (the oldest and largest museum in India). Kolkata’s streets are dotted with old mansions and buildings that tell stories of its colonial past. Walking through the city can feel like stepping back into a different era. Finally, Kolkata is also known for its street food.”

ICPR 2024 followed a two-round paper submission format. We received a total of 2135 papers (1501 papers in round-1 submissions, and 634 papers in round-2 submissions). Each paper, on average, received 2.84 reviews, in single-blind mode. For the first-round papers we had a rebuttal option available to authors.

In total, 945 papers (669 from round-1 and 276 from round-2) were accepted for presentation, resulting in an acceptance rate of 44.26%, which is consistent with previous ICPR events. At ICPR 2024 the papers were categorized into six tracks: Artificial Intelligence, Machine Learning for Pattern Analysis; Computer Vision and Robotic Perception; Image, Video, Speech, and Signal Analysis; Biometrics and Human-Machine Interaction; Document and Media Analysis; and Biomedical Image Analysis and Informatics.

The main conference ran over December 2–5, 2024. The main program included the presentation of 188 oral papers (19.89% of the accepted papers), 757 poster papers and 12 competition papers (out of 15 submitted). A total 10 oral sessions were held concurrently in four meeting rooms with a total of 40 oral sessions. In total 24 workshops and 7 tutorials were held on December 1, 2024.

The plenary sessions included three prize lectures and three invited presentations. The prize lectures were delivered by Tin Kam Ho (IBM Research, USA; King Sun



Fu Prize winner), Xiaolong Wang (University of California, San Diego, USA; J.K. Aggarwal Prize winner), and Guoying Zhao (University of Oulu, Finland; Maria Petrou Prize winner). The invited speakers were Timothy Hospedales (University of Edinburgh, UK), Venu Govindaraju (University at Buffalo, USA), and Shuicheng Yan (Skywork AI, Singapore).

Several best paper awards were presented in ICPR: the Piero Zamperoni Award for the best paper authored by a student, the BIRPA Best Industry Related Paper Award, and the Best Paper Awards and Best Student Paper Awards for each of the six tracks of ICPR 2024.

The organization of such a large conference would not be possible without the help of many volunteers. Our special gratitude goes to the Program Chairs (Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa and Cheng-Lin Liu), for their leadership in organizing the program. Thanks to our Publication Chairs (Ananda S. Chowdhury and Wataru Ohyama) for handling the overwhelming workload of publishing the conference proceedings. We also thank our Competition Chairs (Richard Zanibbi, Lianwen Jin and Laurence Likforman-Sulem) for arranging 12 important competitions as part of ICPR 2024. We are thankful to our Workshop Chairs (P. Shivakumara, Stephanie Schuckers, Jean-Marc Ogier and Prabir Bhattacharya) and Tutorial Chairs (B.B. Chaudhuri, Michael R. Jenkin and Guoying Zhao) for arranging the workshops and tutorials on emerging topics. ICPR 2024, for the first time, held a Doctoral Consortium. We would like to thank our Doctoral Consortium Chairs (Véronique Eglin, Dan Lopresti and Mayank Vatsa) for organizing it.

Thanks go to the Track Chairs and the meta reviewers who devoted significant time to the review process and preparation of the program. We also sincerely thank the reviewers who provided valuable feedback to the authors.

Finally, we acknowledge the work of other conference committee members, like the Organizing Chairs and Organizing Committee Members, Finance Chairs, Award Chair, Sponsorship Chairs, and Exhibition and Demonstration Chairs, Visa Chair, Publicity Chairs, and Women in ICPR Chairs, whose efforts made this event successful. We also thank our event manager Alpcord Network for their help.

We hope that all the participants found the technical program informative and enjoyed the sights, culture and cuisine of Kolkata.

October 2024

Umapada Pal  
Josef Kittler  
Anil Jain

# Organization

## General Chairs

Umapada Pal  
Josef Kittler  
Anil Jain

Indian Statistical Institute, Kolkata, India  
University of Surrey, UK  
Michigan State University, USA

## Program Chairs

Apostolos Antonacopoulos  
Subhasis Chaudhuri  
Rama Chellappa  
Cheng-Lin Liu

University of Salford, UK  
Indian Institute of Technology, Bombay, India  
Johns Hopkins University, USA  
Institute of Automation, Chinese Academy of  
Sciences, China

## Publication Chairs

Ananda S. Chowdhury  
Wataru Ohyama

Jadavpur University, India  
Tokyo Denki University, Japan

## Competition Chairs

Richard Zanibbi  
Lianwen Jin  
Laurence Likforman-Sulem

Rochester Institute of Technology, USA  
South China University of Technology, China  
Télécom Paris, France

## Workshop Chairs

P. Shivakumara  
Stephanie Schuckers  
Jean-Marc Ogier  
Prabir Bhattacharya

University of Salford, UK  
Clarkson University, USA  
Université de la Rochelle, France  
Concordia University, Canada

## **Tutorial Chairs**

B. B. Chaudhuri	Indian Statistical Institute, Kolkata, India
Michael R. Jenkin	York University, Canada
Guoying Zhao	University of Oulu, Finland

## **Doctoral Consortium Chairs**

Véronique Eglin	CNRS, France
Daniel P. Lopresti	Lehigh University, USA
Mayank Vatsa	Indian Institute of Technology, Jodhpur, India

## **Organizing Chairs**

Saumik Bhattacharya	Indian Institute of Technology, Kharagpur, India
Palash Ghosal	Sikkim Manipal University, India

## **Organizing Committee**

Santanu Phadikar	West Bengal University of Technology, India
SK Md Obaidullah	Aliah University, India
Sayantari Ghosh	National Institute of Technology Durgapur, India
Himadri Mukherjee	West Bengal State University, India
Nilamadhaba Tripathy	Clarivate Analytics, USA
Chayan Halder	West Bengal State University, India
Shibaprasad Sen	Techno Main Salt Lake, India

## **Finance Chairs**

Kaushik Roy	West Bengal State University, India
Michael Blumenstein	University of Technology Sydney, Australia

## **Awards Committee Chair**

Arpan Pal	Tata Consultancy Services, India
-----------	----------------------------------

## Sponsorship Chairs

P. J. Narayanan	Indian Institute of Technology, Hyderabad, India
Yasushi Yagi	Osaka University, Japan
Venu Govindaraju	University at Buffalo, USA
Alberto Bel Bimbo	Università di Firenze, Italy

## Exhibition and Demonstration Chairs

Arjun Jain	FastCode AI, India
Agnimitra Biswas	National Institute of Technology, Silchar, India

## International Liaison, Visa Chair

Balasubramanian Raman	Indian Institute of Technology, Roorkee, India
-----------------------	--

## Publicity Chairs

Dipti Prasad Mukherjee	Indian Statistical Institute, Kolkata, India
Bob Fisher	University of Edinburgh, UK
Xiaojun Wu	Jiangnan University, China

## Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian Statistical Institute, Kolkata, India

## Event Manager

Alpcord Network

## **Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis**

Larry O’Gorman	Nokia Bell Labs, USA
Dacheng Tao	University of Sydney, Australia
Petia Radeva	University of Barcelona, Spain
Susmita Mitra	Indian Statistical Institute, Kolkata, India
Jiliang Tang	Michigan State University, USA

## **Track Chairs – Computer and Robot Vision**

C. V. Jawahar	International Institute of Information Technology (IIIT), Hyderabad, India
João Paulo Papa	São Paulo State University, Brazil
Maja Pantic	Imperial College London, UK
Gang Hua	Dolby Laboratories, USA
Junwei Han	Northwestern Polytechnical University, China

## **Track Chairs – Image, Speech, Signal and Video Processing**

P. K. Biswas	Indian Institute of Technology, Kharagpur, India
Shang-Hong Lai	National Tsing Hua University, Taiwan
Hugo Jair Escalante	INAOE, CINVESTAV, Mexico
Sergio Escalera	Universitat de Barcelona, Spain
Prem Natarajan	University of Southern California, USA

## **Track Chairs – Biometrics and Human Computer Interaction**

Richa Singh	Indian Institute of Technology, Jodhpur, India
Massimo Tistarelli	University of Sassari, Italy
Vishal Patel	Johns Hopkins University, USA
Wei-Shi Zheng	Sun Yat-sen University, China
Jian Wang	Snap, USA

## Track Chairs – Document Analysis and Recognition

Xiang Bai	Huazhong University of Science and Technology, China
David Doermann	University at Buffalo, USA
Josep Lladós	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

## Track Chairs – Biomedical Imaging and Bioinformatics

Jayanta Mukhopadhyay	Indian Institute of Technology, Kharagpur, India
Xiaoyi Jiang	Universität Münster, Germany
Seong-Whan Lee	Korea University, Korea

## Metareviewers (Conference Papers and Competition Papers)

Wael Abd-Almageed	University of Southern California, USA
Maya Aghaei	NHL Stenden University, Netherlands
Alireza Alaei	Southern Cross University, Australia
Rajagopalan N. Ambasamudram	Indian Institute of Technology, Madras, India
Suyash P. Awate	Indian Institute of Technology, Bombay, India
Inci M. Baytas	Bogazici University, Turkey
Aparna Bharati	Lehigh University, USA
Brojeshwar Bhowmick	Tata Consultancy Services, India
Jean-Christophe Burie	University of La Rochelle, France
Gustavo Carneiro	University of Surrey, UK
Chee Seng Chan	Universiti Malaya, Malaysia
Sumohana S. Channappayya	Indian Institute of Technology, Hyderabad, India
Dongdong Chen	Microsoft, USA
Shengyong Chen	Tianjin University of Technology, China
Jun Cheng	Institute for Infocomm Research, A*STAR, Singapore
Albert Clapés	University of Barcelona, Spain
Oscar Dalmau	Center for Research in Mathematics, Mexico

Tyler Derr	Vanderbilt University, USA
Abhinav Dhall	Indian Institute of Technology, Ropar, India
Bo Du	Wuhan University, China
Yuxuan Du	University of Sydney, Australia
Ayman S. El-Baz	University of Louisville, USA
Francisco Escolano	University of Alicante, Spain
Siamac Fazli	Nazarbayev University, Kazakhstan
Jianjiang Feng	Tsinghua University, China
Gernot A. Fink	TU Dortmund University, Germany
Alicia Fornes	CVC, Spain
Junbin Gao	University of Sydney, Australia
Yan Gao	Amazon, USA
Yongsheng Gao	Griffith University, Australia
Caren Han	University of Melbourne, Australia
Ran He	Institute of Automation, Chinese Academy of Sciences, China
Tin Kam Ho	IBM, USA
Di Huang	Beihang University, China
Kaizhu Huang	Duke Kunshan University, China
Donato Impedovo	University of Bari, Italy
Julio Jacques	University of Barcelona and Computer Vision Center, Spain
Lianwen Jin	South China University of Technology, China
Wei Jin	Emory University, USA
Danilo Samuel Jodas	São Paulo State University, Brazil
Manjunath V. Joshi	DA-IICT, India
Jayashree Kalpathy-Cramer	Massachusetts General Hospital, USA
Dimosthenis Karatzas	Computer Vision Centre, Spain
Hamid Karimi	Utah State University, USA
Baiying Lei	Shenzhen University, China
Guoqi Li	Chinese Academy of Sciences, and Peng Cheng Lab, China
Laurence Likforman-Sulem	Institut Polytechnique de Paris/Télécom Paris, France
Aishan Liu	Beihang University, China
Bo Liu	Bytedance, USA
Chen Liu	Clarkson University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China
Hongmin Liu	University of Science and Technology Beijing, China
Hui Liu	Michigan State University, USA

Jing Liu	Institute of Automation, Chinese Academy of Sciences, China
Li Liu	University of Oulu, Finland
Qingshan Liu	Nanjing University of Posts and Telecommunications, China
Adrian P. Lopez-Monroy	Centro de Investigacion en Matematicas AC, Mexico
Daniel P. Lopresti	Lehigh University, USA
Shijian Lu	Nanyang Technological University, Singapore
Yong Luo	Wuhan University, China
Andreas K. Maier	FAU Erlangen-Nuremberg, Germany
Davide Maltoni	University of Bologna, Italy
Hong Man	Stevens Institute of Technology, USA
Lingtong Min	Northwestern Polytechnical University, China
Paolo Napoletano	University of Milano-Bicocca, Italy
Kamal Nasrollahi	Milestone Systems, Aalborg University, Denmark
Marcos Ortega	University of A Coruña, Spain
Shivakumara Palaiahnakote	University of Salford, UK
P. Jonathon Phillips	NIST, USA
Filiberto Pla	University Jaume I, Spain
Ajit Rajwade	Indian Institute of Technology, Bombay, India
Shanmuganathan Raman	Indian Institute of Technology, Gandhinagar, India
Imran Razzak	UNSW, Australia
Beatriz Remeseiro	University of Oviedo, Spain
Gustavo Rohde	University of Virginia, USA
Partha Pratim Roy	Indian Institute of Technology, Roorkee, India
Sanjoy K. Saha	Jadavpur University, India
Joan Andreu Sánchez	Universitat Politècnica de València, Spain
Claudio F. Santos	UFSCar, Brazil
Shin'ichi Satoh	National Institute of Informatics, Japan
Stephanie Schuckers	Clarkson University, USA
Srirangaraj Setlur	University at Buffalo, SUNY, USA
Debdoot Sheet	Indian Institute of Technology, Kharagpur, India
Jun Shen	University of Wollongong, Australia
Li Shen	JD Explore Academy, China
Chen Shengyong	Zhejiang University of technology and Tianjin University of Technology, China
Andy Song	RMIT University, Australia
Akihiro Sugimoto	National Institute of Informatics, Japan
Qianru Sun	Singapore Management University, Singapore
Arijit Sur	Indian Institute of Technology, Guwahati, India
Estefania Talavera	University of Twente, Netherlands



Wei Tang	University of Illinois at Chicago, USA
Joao M. Tavares	Universidade do Porto, Portugal
Jun Wan	NLPR, CASIA, China
Le Wang	Xi'an Jiaotong University, China
Lei Wang	Australian National University, Australia
Xiaoyang Wang	Tencent AI Lab, USA
Xinggang Wang	Huazhong University of Science and Technology, China
Xiao-Jun Wu	Jiangnan University, China
Yiding Yang	Bytedance, China
Xiwen Yao	Northwestern Polytechnical University, China
Xu-Cheng Yin	University of Science and Technology Beijing, China
Baosheng Yu	University of Sydney, Australia
Shiqi Yu	Southern University of Science and Technology, China
Xin Yuan	Westlake University, China
Yibing Zhan	JD Explore Academy, China
Jing Zhang	University of Sydney, Australia
Lefei Zhang	Wuhan University, China
Min-Ling Zhang	Southeast University, China
Wenbin Zhang	Florida International University, USA
Jiahuan Zhou	Peking University, China
Sanping Zhou	Xi'an Jiaotong University, China
Tianyi Zhou	University of Maryland, USA
Lei Zhu	Shandong Normal University, China
Pengfei Zhu	Tianjin University, China
Wangmeng Zuo	Harbin Institute of Technology, China

## **Reviewers (Competition Papers)**

Liangcai Gao	Da-Han Wang
Mingxin Huang	Yang Xue
Lei Kang	Wentao Yang
Wenhui Liao	Jiixin Zhang
Yuliang Liu	Yiwu Zhong
Yongxin Shi	

## Reviewers (Conference Papers)

Aakanksha Aakanksha  
 Aayush Singla  
 Abdul Muqet  
 Abhay Yadav  
 Abhijeet Vijay Nandedkar  
 Abhimanyu Sahu  
 Abhinav Rajvanshi  
 Abhisek Ray  
 Abhishek Shrivastava  
 Abhra Chaudhuri  
 Aditi Roy  
 Adriano Simonetto  
 Adrien Maglo  
 Ahmed Abdulkadir  
 Ahmed Boudissa  
 Ahmed Hamdi  
 Ahmed Rida Sekkat  
 Ahmed Sharafeldeen  
 Aiman Farooq  
 Aishwarya Venkataramanan  
 Ajay Kumar  
 Ajay Kumar Reddy Poreddy  
 Ajita Rattani  
 Ajoy Mondal  
 Akbar K.  
 Akbar Telikani  
 Akshay Agarwal  
 Akshit Jindal  
 Al Zadid Sultan Bin Habib  
 Albert Clapés  
 Alceu Britto  
 Alejandro Peña  
 Alessandro Ortis  
 Alessia Auriemma Citarella  
 Alexandre Stenger  
 Alexandros Sopasakis  
 Alexia Toumpa  
 Ali Khan  
 Alik Pramanick  
 Alireza Alaei  
 Alper Yilmaz  
 Aman Verma  
 Amit Bhardwaj

Amit More  
 Amit Nandedkar  
 Amitava Chatterjee  
 Amos L. Abbott  
 Amrita Mohan  
 Anand Mishra  
 Ananda S. Chowdhury  
 Anastasia Zakharova  
 Anastasios L. Kesidis  
 Andras Horvath  
 Andre Gustavo Hochuli  
 André P. Kelm  
 Andre Wyzykowski  
 Andrea Bottino  
 Andrea Lagorio  
 Andrea Torsello  
 Andreas Fischer  
 Andreas K. Maier  
 Andreu Girbau Xalabarder  
 Andrew Beng Jin Teoh  
 Andrew Shin  
 Andy J. Ma  
 Aneesh S. Chivukula  
 Ángela Casado-García  
 Anh Quoc Nguyen  
 Anindya Sen  
 Anirban Saha  
 Anjali Gautam  
 Ankan Bhattacharyya  
 Ankit Jha  
 Anna Scius-Bertrand  
 Annalisa Franco  
 Antoine Doucet  
 Antonino Staiano  
 Antonio Fernández  
 Antonio Parziale  
 Anu Singha  
 Anustup Choudhury  
 Anwesan Pal  
 Anwasha Sengupta  
 Archisman Adhikary  
 Arjan Kuijper  
 Arnab Kumar Das

Arnav Bhavsar  
Arnav Varma  
Arpita Dutta  
Arshad Jamal  
Artur Jordao  
Arunkumar Chinnaswamy  
Aryan Jadon  
Aryaz Baradarani  
Ashima Anand  
Ashis Dhara  
Ashish Phophalia  
Ashok K. Bhateja  
Ashutosh Vaish  
Ashwani Kumar  
Asifuzzaman Lasker  
Atefeh Khoshkhahtinat  
Athira Nambiar  
Attilio Fiandrotti  
Avandra S. Hemachandra  
Avik Hati  
Avinash Sharma  
B. H. Shekar  
B. Uma Shankar  
Bala Krishna Thunakala  
Balaji Tk  
Balázs Pálffy  
Banafsheh Adami  
Bang-Dang Pham  
Baochang Zhang  
Baodi Liu  
Bashirul Azam Biswas  
Beiduo Chen  
Benedikt Kottler  
Beomseok Oh  
Berkay Aydin  
Berlin S. Shaheema  
Bertrand Kerautret  
Bettina Finzel  
Bhavana Singh  
Bibhas C. Dhara  
Bilge Günsel  
Bin Chen  
Bin Li  
Bin Liu  
Bin Yao  
Bin-Bin Jia  
Binbin Yong  
Bindita Chaudhuri  
Bindu Madhavi Tummala  
Binh M. Le  
Bi-Ru Dai  
Bo Huang  
Bo Jiang  
Bob Zhang  
Bowen Liu  
Bowen Zhang  
Boyang Zhang  
Boyu Diao  
Boyun Li  
Brian M. Sadler  
Bruce A. Maxwell  
Bryan Bo Cao  
Buddhika L. Semage  
Bushra Jalil  
Byeong-Seok Shin  
Byung-Gyu Kim  
Caihua Liu  
Cairong Zhao  
Camille Kurtz  
Carlos A. Caetano  
Carlos D. Martá-Nez-Hinarejos  
Ce Wang  
Cevahir Cigla  
Chakravarthy Bhagvati  
Chandrakanth Vipparla  
Changchun Zhang  
Changde Du  
Changkun Ye  
Changxu Cheng  
Chao Fan  
Chao Guo  
Chao Qu  
Chao Wen  
Chayan Halder  
Che-Jui Chang  
Chen Feng  
Chenan Wang  
Cheng Yu  
Chenghao Qian  
Cheng-Lin Liu

Chengxu Liu  
Chenru Jiang  
Chensheng Peng  
Chetan Ralekar  
Chih-Wei Lin  
Chih-Yi Chiu  
Chinmay Sahu  
Chintan Patel  
Chintan Shah  
Chiranjoy Chattopadhyay  
Chong Wang  
Choudhary Shyam Prakash  
Christophe Charrier  
Christos Smailis  
Chuanwei Zhou  
Chun-Ming Tsai  
Chunpeng Wang  
Ciro Russo  
Claudio De Stefano  
Claudio F. Santos  
Claudio Marrocco  
Connor Levenson  
Constantine Dovrolis  
Constantine Kotropoulos  
Dai Shi  
Dakshina Ranjan Kisku  
Dan Anitei  
Dandan Zhu  
Daniela Pamplona  
Danli Wang  
Danqing Huang  
Daoan Zhang  
Daqing Hou  
David A. Clausi  
David Freire Obregon  
David Münch  
David Pujol Perich  
Davide Marelli  
De Zhang  
Debalina Barik  
Debapriya Roy (Kundu)  
Debashis Das  
Debashis Das Chakladar  
Debi Prosad Dogra  
Debraj D. Basu  
Decheng Liu  
Deen Dayal Mohan  
Deep A. Patel  
Deepak Kumar  
Dengpan Liu  
Denis Coquenat  
Désiré Sidibé  
Devesh Walawalkar  
Dewan Md. Farid  
Di Ming  
Di Qiu  
Di Yuan  
Dian Jia  
Dianmo Sheng  
Diego Thomas  
Diganta Saha  
Dimitri Bulatov  
Dimpy Varshni  
Dingcheng Yang  
Dipanjan Das  
Dipanjoyoti Paul  
Divya Biligere Shivanna  
Divya Saxena  
Divya Sharma  
Dmitrii Matveichev  
Dmitry Minskiy  
Dmitry V. Sorokin  
Dong Zhang  
Donghua Wang  
Donglin Zhang  
Dongming Wu  
Dongqiangzi Ye  
Dongqing Zou  
Dongrui Liu  
Dongyang Zhang  
Dongzhan Zhou  
Douglas Rodrigues  
Duarte Folgado  
Duc Minh Vo  
Duoxuan Pei  
Durai Arun Pannir Selvam  
Durga Bhavani S.  
Eckart Michaelsen  
Elena Goyanes  
Élodie Puybareau

Emanuele Vivoli  
Emna Ghorbel  
Enrique Naredo  
Enyu Cai  
Eric Patterson  
Ernest Valveny  
Eva Blanco-Mallo  
Eva Breznik  
Evangelos Sartinas  
Fabio Solari  
Fabiola De Marco  
Fan Wang  
Fangda Li  
Fangyuan Lei  
Fangzhou Lin  
Fangzhou Luo  
Fares Bougourzi  
Farman Ali  
Fatiha Mokdad  
Fei Shen  
Fei Teng  
Fei Zhu  
Feiyan Hu  
Felipe Gomes Oliveira  
Feng Li  
Fengbei Liu  
Fenghua Zhu  
Fillipe D. M. De Souza  
Flavio Piccoli  
Flavio Prieto  
Florian Kleber  
Francesc Serratosa  
Francesco Bianconi  
Francesco Castro  
Francesco Ponzio  
Francisco Javier Hernández López  
Frédéric Rayar  
Furkan Osman Kar  
Fushuo Huo  
Fuxiao Liu  
Fu-Zhao Ou  
Gabriel Turinici  
Gabrielle Flood  
Gajjala Viswanatha Reddy  
Gaku Nakano  
Galal Binamakhshen  
Ganesh Krishnasamy  
Gang Pan  
Gangyan Zeng  
Gani Rahmon  
Gaurav Harit  
Gennaro Vessio  
Genoveffa Tortora  
George Azzopardi  
Gerard Ortega  
Gerardo E. Altamirano-Gomez  
Gernot A. Fink  
Gibran Benitez-Garcia  
Gil Ben-Artzi  
Gilbert Lim  
Giorgia Minello  
Giorgio Fumera  
Giovanna Castellano  
Giovanni Puglisi  
Giulia Orrù  
Giuliana Ramella  
Gökçe Uludoğan  
Gopi Ramena  
Gorthi Rama Krishna Sai Subrahmanyam  
Gourav Datta  
Gowri Srinivasa  
Gozde Sahin  
Gregory Randall  
Guanjie Huang  
Guanjun Li  
Guanwen Zhang  
Guanyu Xu  
Guanyu Yang  
Guanzhou Ke  
Guhnoo Yun  
Guido Borghi  
Guilherme Brandão Martins  
Guillaume Caron  
Guillaume Tochon  
Guocai Du  
Guohao Li  
Guoqiang Zhong  
Guorong Li  
Guotao Li  
Gurman Gill

Haechang Lee  
Haichao Zhang  
Haidong Xie  
Haifeng Zhao  
Haimei Zhao  
Hainan Cui  
Haixia Wang  
Haiyan Guo  
Hakime Ozturk  
Hamid Kazemi  
Han Gao  
Hang Zou  
Hanjia Lyu  
Hanjoo Cho  
Hanqing Zhao  
Hanyuan Liu  
Hanzhou Wu  
Hao Li  
Hao Meng  
Hao Sun  
Hao Wang  
Hao Xing  
Hao Zhao  
Haoan Feng  
Haodi Feng  
Haofeng Li  
Haoji Hu  
Haojie Hao  
Haojun Ai  
Haopeng Zhang  
Haoran Li  
Haoran Wang  
Haorui Ji  
Haoxiang Ma  
Haoyu Chen  
Haoyue Shi  
Harald Koestler  
Harbinder Singh  
Harris V. Georgiou  
Hasan F. Ates  
Hasan S. M. Al-Khaffaf  
Hatef Otroschi Shahreza  
Hebeizi Li  
Heng Zhang  
Hengli Wang  
Hengyue Liu  
Hertog Nugroho  
Hieyong Jeong  
Himadri Mukherjee  
Hoai Ngo  
Hoda Mohaghegh  
Hong Liu  
Hong Man  
Hongcheng Wang  
Hongjian Zhan  
Hongxi Wei  
Hongyu Hu  
Hoseong Kim  
Hossein Ebrahimnezhad  
Hossein Malekmohamadi  
Hrishav Bakul Barua  
Hsueh-Yi Sean Lin  
Hua Wei  
Huafeng Li  
Huali Xu  
Huaming Chen  
Huan Wang  
Huang Chen  
Huanran Chen  
Hua-Wen Chang  
Huawen Liu  
Huayi Zhan  
Hugo Jair Escalante  
Hui Chen  
Hui Li  
Huichen Yang  
Huiqiang Jiang  
Huiyuan Yang  
Huizi Yu  
Hung T. Nguyen  
Hyeongyu Kim  
Hyeonjeong Park  
Hyeonjun Lee  
Hymalai Bello  
Hyung-Gun Chi  
Hyunsoo Kim  
I-Chen Lin  
Ik Hyun Lee  
Ilan Shimshoni  
Imad Eddine Toubal

Imran Sarker  
Inderjot Singh Saggu  
Indrani Mukherjee  
Indranil Sur  
Ines Rieger  
Ioannis Pierros  
Irina Rabaev  
Ivan V. Medri  
J. Rafid Siddiqui  
Jacek Komorowski  
Jacopo Bonato  
Jacson Rodrigues Correia-Silva  
Jaekoo Lee  
Jaime Cardoso  
Jakob Gawlikowski  
Jakub Nalepa  
James L. Wayman  
Jan Čech  
Jangho Lee  
Jani Boutellier  
Javier Gurrola-Ramos  
Javier Lorenzo-Navarro  
Jayasree Saha  
Jean Lee  
Jean Paul Barddal  
Jean-Bernard Hayet  
Jean-Philippe G. Tarel  
Jean-Yves Ramel  
Jenny Benois-Pineau  
Jens Bayer  
Jerin Geo James  
Jesús Miguel García-Gorrostieta  
Jia Qu  
Jiahong Chen  
Jiaji Wang  
Jian Hou  
Jian Liang  
Jian Xu  
Jian Zhu  
Jianfeng Lu  
Jianfeng Ren  
Jiangfan Liu  
Jianguo Wang  
Jiangyan Yi  
Jiangyong Duan  
Jianhua Yang  
Jianhua Zhang  
Jianhui Chen  
Jianjia Wang  
Jianli Xiao  
Jianqiang Xiao  
Jianwu Wang  
Jianxin Zhang  
Jianxiong Gao  
Jianxiong Zhou  
Jianyu Wang  
Jianzhong Wang  
Jiaru Zhang  
Jiashu Liao  
Jiaxin Chen  
Jiaxin Lu  
Jiaxing Ye  
Jiaxuan Chen  
Jiaxuan Li  
Jiayi He  
Jiayin Lin  
Jie Ou  
Jiehua Zhang  
Jiejie Zhao  
Jignesh S. Bhatt  
Jin Gao  
Jin Hou  
Jin Hu  
Jin Shang  
Jing Tian  
Jing Yu Chen  
Jingfeng Yao  
Jinglun Feng  
Jingtong Yue  
Jingwei Guo  
Jingwen Xu  
Jingyuan Xia  
Jingzhe Ma  
Jinhong Wang  
Jinjia Wang  
Jinlai Zhang  
Jinlong Fan  
Jinming Su  
Jinrong He  
Jintao Huang

Jinwoo Ahn  
Jinwoo Choi  
Jinyang Liu  
Jinyu Tian  
Jionghao Lin  
Jiuding Duan  
Jiwei Shen  
Jiyang Pan  
Jiyoun Kim  
João Papa  
Johan Debayle  
John Atanbori  
John Wilson  
John Zhang  
Jónathan Heras  
Joohi Chauhan  
Jorge Calvo-Zaragoza  
Jorge Figueroa  
Jorma Laaksonen  
José Joaquim De Moura Ramos  
Jose Vicent  
Joseph Damilola Akinyemi  
Josiane Zerubia  
Juan Wen  
Judit Szücs  
Juepeng Zheng  
Juha Roning  
Jumana H. Alsubhi  
Jun Cheng  
Jun Ni  
Jun Wan  
Junghyun Cho  
Junjie Liang  
Junjie Ye  
Junlin Hu  
Juntong Ni  
Junxin Lu  
Junxuan Li  
Junyaup Kim  
Junyeong Kim  
Jürgen Seiler  
Jushang Qiu  
Juyang Weng  
Jyostna Devi Bodapati  
Jyoti Singh Kirar  
Kai Jiang  
Kaiqiang Song  
Kalidas Yeturu  
Kalle Åström  
Kamalakar Vijay Thakare  
Kang Gu  
Kang Ma  
Kanji Tanaka  
Karthik Seemakurthy  
Kaushik Roy  
Kavisha Jayathunge  
Kazuki Uehara  
Ke Shi  
Keigo Kimura  
Keiji Yanai  
Kelton A. P. Costa  
Kenneth Camilleri  
Kenny Davila  
Ketan Atul Bapat  
Ketan Kotwal  
Kevin Desai  
Keyu Long  
Khadiga Mohamed Ali  
Khakon Das  
Khan Muhammad  
Kilho Son  
Kim-Ngan Nguyen  
Kishan Kc  
Kishor P. Upla  
Klaas Dijkstra  
Komal Bharti  
Konstantinos Triaridis  
Kostas Ioannidis  
Koyel Ghosh  
Kripabandhu Ghosh  
Krishnendu Ghosh  
Kshitij S. Jadhav  
Kuan Yan  
Kun Ding  
Kun Xia  
Kun Zeng  
Kunal Banerjee  
Kunal Biswas  
Kunchi Li  
Kurban Ubul



Lahiru N. Wijayasingha  
Laines Schmalwasser  
Lakshman Mahto  
Lala Shakti Swarup Ray  
Lale Akarun  
Lan Yan  
Lawrence Amadi  
Lee Kang Il  
Lei Fan  
Lei Shi  
Lei Wang  
Leonardo Rossi  
Lequan Lin  
Levente Tamas  
Li Bing  
Li Li  
Li Ma  
Li Song  
Lia Morra  
Liang Xie  
Liang Zhao  
Lianwen Jin  
Libing Zeng  
Lidia Sánchez-González  
Lidong Zeng  
Lijun Li  
Likang Wang  
Lili Zhao  
Lin Chen  
Lin Huang  
Linfei Wang  
Ling Lo  
Lingchen Meng  
Lingheng Meng  
Lingxiao Li  
Lingzhong Fan  
Liqi Yan  
Liqiang Jing  
Lisa Gutzeit  
Liu Ziyi  
Liushuai Shi  
Liviú-Daniel Stefan  
Liyuan Ma  
Liyun Zhu  
Lizuo Jin

Longteng Guo  
Lorena Álvarez Rodríguez  
Lorenzo Putzu  
Lu Leng  
Lu Pang  
Lu Wang  
Luan Pham  
Luc Brun  
Luca Guarnera  
Luca Piano  
Lucas Alexandre Ramos  
Lucas Goncalves  
Lucas M. Gago  
Luigi Celona  
Luis C. S. Afonso  
Luis Gerardo De La Fraga  
Luis S. Luevano  
Luis Teixeira  
Lunke Fei  
M. Hassaballah  
Maddimsetti Srinivas  
Mahendran N.  
Mahesh Mohan M. R.  
Maiko Lie  
Mainak Singha  
Makoto Hirose  
Malay Bhattacharyya  
Mamadou Dian Bah  
Man Yao  
Manali J. Patel  
Manav Prabhakar  
Manikandan V. M.  
Manish Bhatt  
Manjunath Shantharamu  
Manuel Curado  
Manuel Günther  
Manuel Marques  
Marc A. Kastner  
Marc Chaumont  
Marc Cheong  
Marc Lalonde  
Marco Cotogni  
Marcos C. Santana  
Mario Molinara  
Mariofanna Milanova

Markus Bauer  
Marlon Becker  
Mårten Wadenbäck  
Martin G. Ljungqvist  
Martin Kämpel  
Martina Pastorino  
Marwan Turki  
Masashi Nishiyama  
Masayuki Tanaka  
Massimo O. Spata  
Matteo Ferrara  
Matthew D. Dawkins  
Matthew Gadd  
Matthew S. Watson  
Maura Pintor  
Max Ehrlich  
Maxim Popov  
Mayukh Das  
Md Baharul Islam  
Md Sajid  
Meghna Kapoor  
Meghna P. Ayyar  
Mei Wang  
Meiqi Wu  
Melissa L. Tijink  
Meng Li  
Meng Liu  
Meng-Luen Wu  
Mengnan Liu  
Mengxi China Guo  
Mengya Han  
Michaël Clément  
Michal Kawulok  
Mickael Coustaty  
Miguel Domingo  
Milind G. Padalkar  
Ming Liu  
Ming Ma  
Mingchen Feng  
Mingde Yao  
Minghao Li  
Mingjie Sun  
Ming-Kuang Daniel Wu  
Mingle Xu  
Mingyong Li  
Mingyuan Jiu  
Minh P. Nguyen  
Minh Q. Tran  
Minheng Ni  
Minsu Kim  
Minyi Zhao  
Mirko Paolo Barbato  
Mo Zhou  
Modesto Castrillón-Santana  
Mohamed Amine Mezghich  
Mohamed Dahmane  
Mohamed Elsharkawy  
Mohamed Yousuf  
Mohammad Hashemi  
Mohammad Khalooei  
Mohammad Khateri  
Mohammad Mahdi Dehshibi  
Mohammad Sadil Khan  
Mohammed Mahmoud  
Moises Diaz  
Monalisha Mahapatra  
Monidipa Das  
Mostafa Kamali Tabrizi  
Mridul Ghosh  
Mrinal Kanti Bhowmik  
Muchao Ye  
Mugalodi Ramesha Rakesh  
Muhammad Rameez Ur Rahman  
Muhammad Suhaib Kanroo  
Muming Zhao  
Munender Varshney  
Munsif Ali  
Na Lv  
Nader Karimi  
Nagabhushan Somraj  
Nakkwan Choi  
Nakul Agarwal  
Nan Pu  
Nan Zhou  
Nancy Mehta  
Nand Kumar Yadav  
Nandakishor Nandakishor  
Nandyala Hemachandra  
Nanfeng Jiang  
Narayan Hegde

Narayan Ji Mishra	Palash Ghosal
Narayan Vetrekar	Pallav Dutta
Narendra D. Londhe	Paolo Rota
Nathalie Girard	Paramanand Chandramouli
Nati Ofir	Paria Mehrani
Naval Kishore Mehta	Parth Agrawal
Nazmul Shahadat	Partha Basuchowdhuri
Neeti Narayan	Patrick Horain
Neha Bhargava	Pavan Kumar
Nemanja Djuric	Pavan Kumar Anasosalu Vasu
Newlin Shebiah R.	Pedro Castro
Ngo Ba Hung	Peipei Li
Nhat-Tan Bui	Peipei Yang
Niaz Ahmad	Peisong Shen
Nick Theisen	Peiyu Li
Nicolas Passat	Peng Li
Nicolas Ragot	Pengfei He
Nicolas Sidere	Pengrui Quan
Nikolaos Mitianoudis	Pengxin Zeng
Nikolas Ebert	Pengyu Yan
Nilah Ravi Nair	Peter Eisert
Nilesh A. Ahuja	Petra Gomez-Krämer
Nilkanta Sahu	Pierrick Bruneau
Nils Murrugarra-Llerena	Ping Cao
Nina S. T. Hirata	Pingping Zhang
Ninad Aithal	Pintu Kumar
Ning Xu	Pooja Kumari
Ningzhi Wang	Pooja Sahani
Niraj Kumar	Prabhu Prasad Dev
Nirmal S. Punjabi	Pradeep Kumar
Nisha Varghese	Pradeep Singh
Norio Tagawa	Pranjal Sahu
Obaidullah Md Sk	Prasun Roy
Oguzhan Ulucan	Prateek Keserwani
Olfa Mechi	Prateek Mittal
Oliver Tüselmann	Praveen Kumar Chandaliya
Orazio Pontorno	Praveen Tirupattur
Oriol Ramos Terrades	Pravin Nair
Osman Akin	Preeti Gopal
Ouadi Beya	Preety Singh
Ozge Mercanoglu Sincan	Prem Shanker Yadav
Pabitra Mitra	Prerana Mukherjee
Padmanabha Reddy Y. C. A.	Prerna A. Mishra
Palaash Agrawal	Prianka Dey
Palaiahnakote Shivakumara	Priyanka Mudgal

Qc Kha Ng  
Qi Li  
Qi Ming  
Qi Wang  
Qi Zuo  
Qian Li  
Qiang Gan  
Qiang He  
Qiang Wu  
Qiangqiang Zhou  
Qianli Zhao  
Qiansen Hong  
Qiao Wang  
Qidong Huang  
Qihua Dong  
Qin Yuke  
Qing Guo  
Qingbei Guo  
Qingchao Zhang  
Qingjie Liu  
Qinhong Yang  
Qiushi Shi  
Qixiang Chen  
Quan Gan  
Quanlong Guan  
Rachit Chhaya  
Radu Tudor Ionescu  
Rafal Zdunek  
Raghavendra Ramachandra  
Rahimul I. Mazumdar  
Rahul Kumar Ray  
Rajib Dutta  
Rajib Ghosh  
Rakesh Kumar  
Rakesh Paul  
Rama Chellappa  
Rami O. Skaik  
Ramon Aranda  
Ran Wei  
Ranga Raju Vatsavai  
Ranganath Krishnan  
Rasha Friji  
Rashmi S.  
Razaib Tariq  
Rémi Giraud  
René Schuster  
Renlong Hang  
Renrong Shao  
Renu Sharma  
Reza Sadeghian  
Richard Zanibbi  
Rimon Elias  
Rishabh Shukla  
Rita Delussu  
Riya Verma  
Robert J. Ravier  
Robert Sablatnig  
Robin Strand  
Rocco Pietrini  
Rocio Diaz Martin  
Rocio Gonzalez-Diaz  
Rohit Venkata Sai Dulam  
Romain Giot  
Romi Banerjee  
Ru Wang  
Ruben Machucho  
Ruddy Théodose  
Ruggero Pintus  
Rui Deng  
Rui P. Paiva  
Rui Zhao  
Ruifan Li  
Ruigang Fu  
Ruikun Li  
Ruirui Li  
Ruixiang Jiang  
Ruwei Jiang  
Rushi Lan  
Rustam Zhumagambetov  
S. Amutha  
S. Divakar Bhat  
Sagar Goyal  
Sahar Siddiqui  
Sahbi Bahroun  
Sai Karthikeya Vemuri  
Saibal Dutta  
Saihui Hou  
Sajad Ahmad Rather  
Saksham Aggarwal  
Sakthi U.

Salimeh Sekeh  
Samar Bouazizi  
Samia Boukir  
Samir F. Harb  
Samit Biswas  
Samrat Mukhopadhyay  
Samriddha Sanyal  
Sandika Biswas  
Sandip Purnapatra  
Sanghyun Jo  
Sangwoo Cho  
Sanjay Kumar  
Sankaran Iyer  
Sanket Biswas  
Santanu Roy  
Santosh D. Pandure  
Santosh Ku Behera  
Santosh Nanabhau Palaskar  
Santosh Prakash Chouhan  
Sarah S. Alotaibi  
Sasanka Katreddi  
Sathyanarayanan N. Aakur  
Saurabh Yadav  
Sayan Rakshit  
Scott McCloskey  
Sebastian Bunda  
Sejuti Rahman  
Selim Aksoy  
Sen Wang  
Seraj A. Mostafa  
Shanmuganathan Raman  
Shao-Yuan Lo  
Shaoyuan Xu  
Sharia Arfin Tanim  
Shehreen Azad  
Sheng Wan  
Shengdong Zhang  
Shengwei Qin  
Shenyuan Gao  
Sherry X. Chen  
Shibaprasad Sen  
Shigeaki Namiki  
Shiguang Liu  
Shijie Ma  
Shikun Li  
Shinichiro Omachi  
Shirley David  
Shishir Shah  
Shiv Ram Dubey  
Shiva Baghel  
Shivanand S. Gornale  
Shogo Sato  
Shotaro Miwa  
Shreya Ghosh  
Shreya Goyal  
Shuai Su  
Shuai Wang  
Shuai Zheng  
Shuaifeng Zhi  
Shuang Qiu  
Shuhei Tarashima  
Shujing Lyu  
Shuliang Wang  
Shun Zhang  
Shunming Li  
Shunxin Wang  
Shuping Zhao  
Shuquan Ye  
Shuwei Huo  
Shuyue Lan  
Shyi-Chyi Cheng  
Si Chen  
Siddarth Ravichandran  
Sihan Chen  
Siladitya Manna  
Silambarasan Elkana Ebinazer  
Simon Benaïchouche  
Simon S. Woo  
Simone Caldarella  
Simone Milani  
Simone Zini  
Sina Lotfian  
Sitao Luan  
Sivaselvan B.  
Siwei Li  
Siwei Wang  
Siwen Luo  
Siyu Chen  
Sk Aziz Ali  
Sk Md Obaidullah

Sneha Shukla	Suraj Kumar Pandey
Snehasis Banerjee	Surendrabikram Thapa
Snehasis Mukherjee	Suresh Sundaram
Snigdha Sen	Sushil Bhattacharjee
Sofia Casarin	Susmita Ghosh
Soheila Farokhi	Swakkhar Shatabda
Soma Bandyopadhyay	Syed Ms Islam
Son Minh Nguyen	Syed Tousiful Haque
Son Xuan Ha	Taegyeong Lee
Sonal Kumar	Taihui Li
Sonam Gupta	Takashi Shibata
Sonam Nahar	Takeshi Oishi
Song Ouyang	Talha Ahmad Siddiqui
Sotiris Kotsiantis	Tanguy Gernot
Souhaila Djaffal	Tangwen Qian
Soumen Biswas	Tanima Bhowmik
Soumen Sinha	Tanpia Tasnim
Soumitri Chattopadhyay	Tao Dai
Souvik Sengupta	Tao Hu
Spiros Kostopoulos	Tao Sun
Sreeraj Ramachandran	Taoran Yi
Sreya Banerjee	Tapan Shah
Srikanta Pal	Taveena Lotey
Srinivas Arukonda	Teng Huang
Stephane A. Guinard	Tengqi Ye
Su O. Ruan	Teresa Alarcon
Subhadip Basu	Tetsuji Ogawa
Subhajit Paul	Thanh Phuong Nguyen
Subhankar Ghosh	Thanh Tuan Nguyen
Subhankar Mishra	Thattapon Surasak
Subhankar Roy	Thibault Napol�on
Subhash Chandra Pal	Thierry Bouwmans
Subhayu Ghosh	Thinh Truong Huynh Nguyen
Sudip Das	Thomas De Min
Sudipta Banerjee	Thomas E. K. Zielke
Suhas Pillai	Thomas Swearingen
Sujit Das	Tianatahina Jimmy Francky Randrianasoa
Sukalpa Chanda	Tianheng Cheng
Sukhendu Das	Tianjiao He
Suklav Ghosh	Tianyi Wei
Suman K. Ghosh	Tianyuan Zhang
Suman Samui	Tianyue Zheng
Sumit Mishra	Tiecheng Song
Sungho Suh	Tilottama Goswami
Sunny Gupta	Tim B�chner

Tim H. Langer	Wataru Ohyama
Tim Raven	Wee Kheng Leow
Ting kai Liu	Wei Chen
Tingting Yao	Wei Cheng
Tobias Meisen	Wei Hua
Toby P. Breckon	Wei Lu
Tong Chen	Wei Pan
Tonghua Su	Wei Tian
Tran Tuan Anh	Wei Wang
Tri-Cong Pham	Wei Wei
Trishna Saikia	Wei Zhou
Trung Quang Truong	Weidi Liu
Tuan T. Nguyen	Weidong Yang
Tuan Vo Van	Weijun Tan
Tushar Shinde	Weimin Lyu
Ujjwal Karn	Weinan Guan
Ukrit Watchareeruetai	Weining Wang
Uma Mudenagudi	Weiqiang Wang
Umarani Jayaraman	Weiwei Guo
V. S. Malemath	Weixia Zhang
Vallidevi Krishnamurthy	Wei-Xuan Bao
Ved Prakash	Weizhong Jiang
Venkata Krishna Kishore Kolli	Wen Xie
Venkata R. Vavilthota	Wenbin Qian
Venkatesh Thirugnana Sambandham	Wenbin Tian
Verónica Maria Vasconcelos	Wenbin Wang
Véronique Ve Eglin	Wenbo Zheng
Víctor E. Alonso-Pérez	Wenhan Luo
Vinay Palakkode	Wenhao Wang
Vinayak S. Nageli	Wen-Hung Liao
Vincent J. Whannou De Dravo	Wenjie Li
Vincenzo Conti	Wenkui Yang
Vincenzo Gattulli	Wenwen Si
Vineet Padmanabhan	Wenwen Yu
Vishakha Pareek	Wenwen Zhang
Viswanath Gopalakrishnan	Wenwu Yang
Vivek Singh Baghel	Wenxi Li
Vivekraj K.	Wenxi Yue
Vladimir V. Arlazarov	Wenxue Cui
Vu-Hoang Tran	Wenzhuo Liu
W. Sylvia Lilly Jebarani	Widhiyo Sudiyono
Wachirawit Ponghiran	Willem Dijkstra
Wafa Khlif	Wolfgang Fuhl
Wang An-Zhi	Xi Zhang
Wanli Xue	Xia Yuan

Xianda Zhang  
Xiang Zhang  
Xiangdong Su  
Xiang-Ru Yu  
Xiangtai Li  
Xiangyu Xu  
Xiao Guo  
Xiao Hu  
Xiao Wu  
Xiao Yang  
Xiaofeng Zhang  
Xiaogang Du  
Xiaoguang Zhao  
Xiaoheng Jiang  
Xiaohong Zhang  
Xiaohua Huang  
Xiaohua Li  
Xiao-Hui Li  
Xiaolong Sun  
Xiaosong Li  
Xiaotian Li  
Xiaoting Wu  
Xiaotong Luo  
Xiaoyan Li  
Xiaoyang Kang  
Xiaoyi Dong  
Xin Guo  
Xin Lin  
Xin Ma  
Xinchi Zhou  
Xingguang Zhang  
Xingjian Leng  
Xingpeng Zhang  
Xingzheng Lyu  
Xinjian Huang  
Xinqi Fan  
Xinqi Liu  
Xinqiao Zhang  
Xinrui Cui  
Xizhan Gao  
Xu Cao  
Xu Ouyang  
Xu Zhao  
Xuan Shen  
Xuan Zhou

Xuchen Li  
Xuejing Lei  
Xuelu Feng  
Xueting Liu  
Xuewei Li  
Xueyi X. Wang  
Xugong Qin  
Xu-Qian Fan  
Xuxu Liu  
Xu-Yao Zhang  
Yan Huang  
Yan Li  
Yan Wang  
Yan Xia  
Yan Zhuang  
Yanan Li  
Yanan Zhang  
Yang Hou  
Yang Jiao  
Yang Liping  
Yang Liu  
Yang Qian  
Yang Yang  
Yang Zhao  
Yangbin Chen  
Yangfan Zhou  
Yanhui Guo  
Yanjia Huang  
Yanjun Zhu  
Yanming Zhang  
Yanqing Shen  
Yaoming Cai  
Yaoxin Zhuo  
Yaoyan Zheng  
Yaping Zhang  
Yaqian Liang  
Yarong Feng  
Yasmina Benmabrouk  
Yasufumi Sakai  
Yasutomo Kawanishi  
Yazeed Alzahrani  
Ye Du  
Ye Duan  
Yechao Zhang  
Yeong-Jun Cho



Yi Huo  
Yi Shi  
Yi Yu  
Yi Zhang  
Yibo Liu  
Yibo Wang  
Yi-Chieh Wu  
Yifan Chen  
Yifei Huang  
Yihao Ding  
Yijie Tang  
Yikun Bai  
Yimin Wen  
Yinan Yang  
Yin-Dong Zheng  
Yinfeng Yu  
Ying Dai  
Yingbo Li  
Yiqiao Li  
Yiqing Huang  
Yisheng Lv  
Yisong Xiao  
Yite Wang  
Yizhe Li  
Yong Wang  
Yonghao Dong  
Yong-Hyuk Moon  
Yongjie Li  
Yongqian Li  
Yongqiang Mao  
Yongxu Liu  
Yongyu Wang  
Yongzhi Li  
Youngha Hwang  
Yousri Kessentini  
Yu Wang  
Yu Zhou  
Yuan Tian  
Yuan Zhang  
Yuanbo Wen  
Yuanxin Wang  
Yubin Hu  
Yubo Huang  
Yuchen Ren  
Yucheng Xing  
Yuchong Yao  
Yuecong Min  
Yuewei Yang  
Yufei Zhang  
Yufeng Yin  
Yugen Yi  
Yuhang Ming  
Yujia Zhang  
Yujun Ma  
Yukiko Kenmochi  
Yun Hoyeoung  
Yun Liu  
Yunhe Feng  
Yunxiao Shi  
Yuru Wang  
Yushun Tang  
Yusuf Osmanlioglu  
Yusuke Fujita  
Yuta Nakashima  
Yuwei Yang  
Yuwu Lu  
Yuxi Liu  
Yuya Obinata  
Yuyao Yan  
Yuzhi Guo  
Zaipeng Xie  
Zander W. Blasingame  
Zedong Wang  
Zeliang Zhang  
Zexin Ji  
Zhanxiang Feng  
Zhaofei Yu  
Zhe Chen  
Zhe Cui  
Zhe Liu  
Zhe Wang  
Zhekun Luo  
Zhen Yang  
Zhenbo Li  
Zhenchun Lei  
Zhenfei Zhang  
Zheng Liu  
Zheng Wang  
Zhengming Yu  
Zhengyin Du

Zhengyun Cheng  
Zhenshen Qu  
Zhenwei Shi  
Zhenzhong Kuang  
Zhi Cai  
Zhi Chen  
Zhibo Chu  
Zhicun Yin  
Zhida Huang  
Zhida Zhang  
Zhifan Gao  
Zhihang Ren  
Zhihang Yuan  
Zhihao Wang  
Zhihua Xie  
Zhihui Wang  
Zhikang Zhang  
Zhiming Zou  
Zhiqi Shao  
Zhiwei Dong  
Zhiwei Qi  
Zhixiang Wang  
Zhixuan Li  
Zhiyu Jiang  
Zhiyuan Yan  
Zhiyuan Yu  
Zhiyuan Zhang  
Zhong Chen  
Zhongwei Teng  
Zhongzhan Huang  
Zhongzhi Yu  
Zhuan Han  
Zhuangzhuang Chen  
Zhuo Liu  
Zhuo Su  
Zhuojun Zou  
Zhuoyue Wang  
Ziang Song  
Zicheng Zhang  
Zied Mnasri  
Zifan Chen  
Žiga Babnik  
Zijing Chen  
Zikai Zhang  
Ziling Huang  
Zilong Du  
Ziqi Cai  
Ziqi Zhou  
Zi-Rui Wang  
Zirui Zhou  
Ziwen He  
Ziyao Zeng  
Ziyi Zhang  
Ziyue Xiang  
Zonglei Jing  
Zongyi Xu

## Contents – Part XV

Supervised Domain Adaptation for Data-Efficient Visible-Infrared Person Re-identification .....	1
<i>Mihir Sahu, Arjun Singh, and Maheshkumar Kolekar</i>	
Anonymisation for Time-Series Human Activity Data .....	17
<i>Tim Hallyburton, Nilah Ravi Nair, Fernando Moya Rueda, René Grzeszick, and Gernot A. Fink</i>	
Representation Biases in Time-Series Human Activity Recognition with Small Sample Sizes .....	33
<i>Nilah Ravi Nair, Lena Schmid, Christopher Reining, Fernando Moya Rueda, Markus Pauly, and Gernot A. Fink</i>	
Secure Sleep Apnea Detection with FHE and Deep Learning on ECG Signals .....	49
<i>Bharat Yalavarthi, Arjun Ramesh Kaushik, Tilak Sharma, Charanjit Jutla, and Nalini Ratha</i>	
Efficient Convolution Operator in FHE Using Summed Area Table .....	65
<i>Bharat Yalavarthi, Charanjit Jutla, and Nalini Ratha</i>	
R-LIME: Rectangular Constraints and Optimization for Local Interpretable Model-agnostic Explanation Methods .....	80
<i>Genji Ohara, Keigo Kimura, and Mineichi Kudo</i>	
Differentially Private Spiking Variational Autoencoder .....	96
<i>Srishti Yadav, Anshul Pundhir, Tanish Goyal, Balasubramanian Raman, and Sanjeev Kumar</i>	
Balancing the Scales: Enhancing Fairness in Facial Emotion Recognition with Latent Alignment .....	113
<i>Syed Sameen Ahmad Rizvi, Aryan Seth, and Pratik Narang</i>	
DDCTrack: Dynamic Token Sampling for Efficient UAV Transformer Tracking .....	129
<i>Guocai Du, Peiyong Zhou, Nurbiya Yadikar, Alimjan Aysa, and Kurban Ubul</i>	

HAPTICS: Human Action Prediction in Real-time via Pose Kinematics . . . . .	145
<i>Niaz Ahmad, Saif Ullah, Jawad Khan, Chanyeok Choi, and Youngmoon Lee</i>	
Predicting the Next Action by Modeling the Abstract Goal . . . . .	162
<i>Debaditya Roy and Basura Fernando</i>	
SHARP: Segmentation of Hands and Arms by Range Using Pseudo-depth for Enhanced Egocentric 3D Hand Pose Estimation and Action Recognition . . .	178
<i>Wiktor Mucha, Michael Wray, and Martin Kampel</i>	
On the Generalization of WiFi-Based Person-Centric Sensing in Through-Wall Scenarios . . . . .	194
<i>Julian Strohmayer and Martin Kampel</i>	
Towards Open-Set Egocentric Action Recognition with Uncertainty Estimation . . . . .	212
<i>Yishan Zou, Christopher Nugent, Matthew Burns, Xiaoming Xi, and Meng Liu</i>	
Temporal Divide-and-Conquer Anomaly Actions Localization in Semi-supervised Videos with Hierarchical Transformer . . . . .	229
<i>Nada Osman and Marwan Torki</i>	
EchoGCN: An Echo Graph Convolutional Network for Skeleton-Based Action Recognition . . . . .	245
<i>Weiwen Qian, Qian Huang, Chang Li, Zhongqi Chen, and Yingchi Mao</i>	
From Category to Scenery: An End-to-End Framework for Multi-person Human-Object Interaction Recognition in Videos . . . . .	262
<i>Tanqiu Qiao, Ruochen Li, Frederick W. B. Li, and Hubert P. H. Shum</i>	
Adaptive Global Gesture Paths and Signature Features for Skeleton-based Gesture Recognition . . . . .	278
<i>Dongzi Shi, Xin Zhang, Jiale Cheng, Tong Xiong, and Hao Ni</i>	
Self-supervised Multi-actor Social Activity Understanding in Streaming Videos . . . . .	293
<i>Shubham Trehan and Sathyanarayanan N. Aakur</i>	
Joint-Temporal Action Segmentation via Multi-action Recognition . . . . .	310
<i>Usfita Kiftiyani and Seungkyu Lee</i>	

Text-Enhanced Zero-Shot Action Recognition: A Training-Free Approach . . . . .	327
<i>Massimo Bosetti, Shibingfeng Zhang, Bendetta Liberatori, Giacomo Zara, Elisa Ricci, and Paolo Rota</i>	
ActNetFormer: Transformer-ResNet Hybrid Method for Semi-supervised Action Recognition in Videos . . . . .	343
<i>Sharana Dharshikgan Suresh Dass, Hrishav Bakul Barua, Ganesh Krishnasamy, Raveendran Paramesran, and Raphaël C.-W. Phan</i>	
RSTAN: Residual Spatio-Temporal Attention Network for End-to-End Human Fall Detection . . . . .	360
<i>Yaru Jiang, Shujing Lyu, Hongjian Zhan, and Yue Lu</i>	
Synthetic Video Generation for Weakly Supervised Cross-Domain Video Anomaly Detection . . . . .	375
<i>Pradeep Narwade, Ryosuke Kawamura, Gaurav Gajbhiye, and Koichiro Niinuma</i>	
Hypergraph Self-Attention and Channel Topology Specialization Network for Automatic Generation of Labanotation . . . . .	392
<i>Weihao Chen, Wanru Xu, and Zhenjiang Miao</i>	
JS-Siamese: Generalized Zero Shot Learning for IMU-based Human Activity Recognition . . . . .	407
<i>Mohammad Al-Saad, Lakshmish Ramaswamy, and Suchendra M. Bhandarkar</i>	
LightHART: Lightweight Human Activity Recognition Transformer . . . . .	425
<i>Syed Tousiful Haque, Jianyuan Ni, Jingcheng Li, Yan Yan, and Anne Hee Hiong Ngu</i>	
Robust Leaf Detection using Shape Priors within Smaller Datasets . . . . .	442
<i>Debojyoti Misra and Tushar Sandhan</i>	
Spectral Aggregation Cross-Square Transformer for Hyperspectral Image Denoising . . . . .	458
<i>Yang Liu, Yantao Ji, Jiahua Xiao, Yu Guo, Peilin Jiang, Haiwei Yang, and Fei Wang</i>	
SDformerFlow: Spiking Neural Network Transformer for Event-based Optical Flow . . . . .	475
<i>Yi Tian and Juan Andrade-Cetto</i>	
<b>Author Index . . . . .</b>	<b>493</b>



# Supervised Domain Adaptation for Data-Efficient Visible-Infrared Person Re-identification

Mihir Sahu<sup>(✉)</sup>, Arjun Singh, and Maheshkumar Kolekar

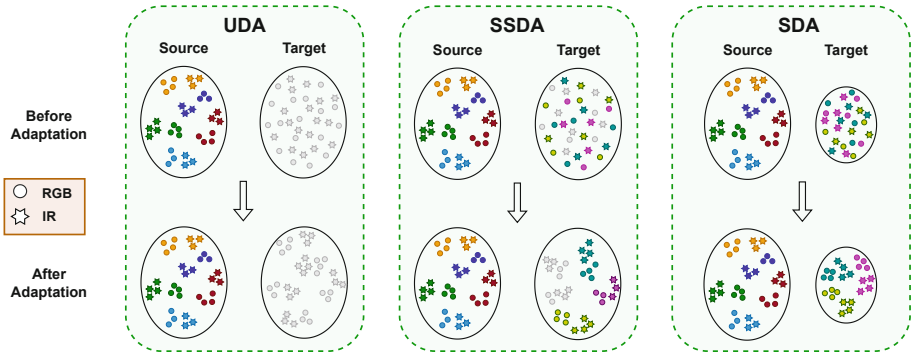
Indian Institute of Technology, Patna, Patna 801106, Bihar, India  
mihirsahu2307@gmail.com, mahesh@iitp.ac.in

**Abstract.** Visible-infrared person re-identification (VI-ReID) is crucial for surveillance and security applications. Several studies have been performed for supervised VI-ReID, and recent methods show excellent retrieval performance on public datasets. However, obtaining VI-ReID data in practical scenarios presents significant challenges due to the necessity for the same identity to be available across different types of cameras, potentially spanning various locations and time frames, along with the arduous task of annotating data owing to modality discrepancies. This motivates us to explore methods requiring limited data from a select number of identities, which is more readily obtainable. To this end, we introduce a novel two-stage learning framework for VI-ReID that efficiently works with scarce data and labels. Our framework focuses on Supervised Domain Adaptation, where a pre-trained model from a source dataset is utilized on a small annotated target dataset. Additionally, we introduce a novel loss, Hetero-Dissimilarity based Maximum Mean Discrepancy (HD-MMD), tailored for adapting heterogeneous source and target domains. Our approach addresses the inherent challenges of domain shift between datasets and modality differences between visible and infrared imagery. Our proposed method outperforms several label-efficient approaches on public VI-ReID datasets while utilizing significantly smaller amount of data. Ablation analysis conducted with several popular baselines reveals the efficacy of our proposed SDA framework and HD-MMD loss in improving retrieval performance. We also demonstrate the ease of integrating our approach with other methods. Code will be released at <https://github.com/Mihirsahu2307/SDA-VI-ReID>.

**Keywords:** VI-ReID · Domain Adaptation · Data-efficient Learning

## 1 Introduction

Visible-infrared person re-identification (VI-ReID) [19, 26, 27] has garnered significant attention in recent years due to its application in 24-hour surveillance and security systems. It involves matching individuals across visible and infrared cameras, presenting unique challenges stemming from the discrepancy between



**Fig. 1.** Comparison of Unsupervised (UDA), Semi-Supervised (SSDA) and Supervised (SDA) Domain Adaptation strategies for VI-ReID. Different colours show different labelled IDs. Grey colour refers to unlabelled data.

the modalities. Supervised VI-ReID [6, 26] has been extensively studied, with some methods achieving more than 90% Rank-1 Retrieval accuracy [3, 6, 7] on publicly available datasets. Recently, to alleviate the issue of tedious manual annotation of VI-ReID datasets, Unsupervised Learning [10, 20, 23] and Unsupervised Domain Adaptation (UDA) [1, 4, 18] based approaches have also been studied which require no annotations, but in their place require copious amounts of data along with huge computational requirements. While substantial progress has been made in developing VI-ReID models, their effectiveness often relies on an abundance of large amounts of paired images of identities in both modalities, termed cross-modal correspondences. This constrains the practical applicability of these approaches as the acquisition of images of the same person in both modalities is a laborious task, as it requires the person to be available at two or more different locations under different lighting conditions and, potentially different times.

To mitigate this issue, we propose a Supervised Domain Adaptation (SDA) based framework for VI-ReID, where a model trained on a large-scale source dataset is transferred to a small annotated target dataset, with no additional unlabelled data. In general, domain adaptation is widely studied in the semi-supervised or unsupervised setting with limited or no target labels for scenarios where gathering the data in abundance might be straightforward, but annotating the data is difficult. But we highlight that for VI-ReID, even garnering large amounts of data in both modalities is an arduous task, thereby resulting in poor performance of the UDA and Semi-Supervised Domain Adaptation (SSDA) based approaches for small target datasets. For instance, an identity present in an RGB camera may not be available in the infrared modality, as this would require changes in lighting conditions and location. This discrepancy could result in a lack of correspondences, adversely affecting training if the data is not collected meticulously. In the proposed setting, very limited data can be collected from a small group of individuals in a controlled fashion. Hence, in

this paper, we explore the SDA setting for VI-ReID. Fig. 1 shows the difference between SDA, SSDA and UDA approaches for VI-ReID. The motivation of the study is inspired by the observations that: 1) In practice, even for unsupervised VI-ReID, amassing datasets with abundant cross-modal correspondences is an onerous task. 2) Compiling and annotating cross-modal data for a small batch of individuals is much easier than gathering huge amounts of unlabelled data with abundant cross-modal correspondences. To this end, we propose a two-stage training procedure for SDA consisting of *Source Pre-training* and *Collaborative Learning*. This framework can be integrated with any baseline method. In the initial Source Pre-training stage, leveraging the annotated source domain, we aim to mitigate the modality gap and learn a robust embedding network. In the Collaborative Learning stage, we train the model using both source and target domains. Further, to bridge the domain gap and distill source knowledge effectively, we propose a Hetero-Dissimilarity based Maximum Mean Discrepancy (HD-MMD) loss. This loss aligns the dissimilarity space of the heterogeneous target domain to that of the heterogeneous source domain. Our study reveals the following insights: 1) Leveraging a large-scale source domain immensely improves the accuracy of the model on the target domain for VI-ReID. 2) Proposed SDA framework can achieve excellent retrieval performance while using as little as 20% of the training data.

In summary, the main contributions of the paper are:

- To address the challenge of collecting large-scale datasets with abundant cross-modal correspondences, we propose a Supervised Domain Adaptation (SDA) setting. SDA leverages a large-scale, annotated source dataset to learn robust representations for a small, annotated target dataset, reducing the dependency on *extensive* cross-modal correspondences and providing a practical solution for VI-ReID.
- We propose a two-stage SDA based framework for VI-ReID that seamlessly integrates with existing VI-ReID methods. This data-efficient framework aims to address the challenges posed by scarcity of data by leveraging the rich annotations of the large-scale source dataset.
- We introduce a novel loss, HD-MMD, which effectively utilizes the scarce cross-modal annotations of the target dataset to learn a robust dissimilarity space for heterogeneous data. This bridges the gap between the heterogeneous source and target domains, which have disjoint label spaces.
- Experiments using the proposed SDA based approach on publicly available datasets demonstrate the retrieval efficacy of our approach over other label-efficient approaches for VI-ReID, while using as little as 20% of the target domain training data. Additionally, we empirically validate the effectiveness of our framework in enhancing the accuracy of various VI-ReID baselines.



## 2 Related Work

### 2.1 Supervised Visible-Infrared Person ReID

Supervised VI-ReID has been extensively studied, with many recent methods achieving excellent results on public VI-ReID datasets. Most works focus on domain-invariant feature learning and reducing the modality discrepancy. Earlier studies [11, 27] focused on metric learning approaches. Some studies also use part-level features to extract local information [8, 12]. Other works focus on explicitly reducing the modality discrepancy by designing suitable loss functions. Jambigi *et al.* [6] introduce a margin-based MMD loss that aligns the modalities at identity level. Recently, Feng *et al.* [3] propose a feature learning paradigm where they erase the shape related features in an attempt to learn other modality-shared discriminative features. However, these methods rely on annotations of large-scale VI-ReID datasets, which limits their practical applicability. Contrary to these methods, we propose to use very limited data of a small set of identities from the target domain to learn a robust embedding network.

### 2.2 Domain Adaptation for Person ReID

Domain Adaptation for Person Re-identification, and consequently, for VI-ReID, is inherently an open set adaptation problem as the label spaces of source and target are mutually exclusive. Most studies focus on UDA for person re-identification. Lee *et al.* [9] introduce a camera-driven curriculum learning framework, wherein they use the camera labels to divide the target dataset into multiple subsets and progressively transfer knowledge from source to target domains. Many studies leverage pseudo labels [2, 17] and learn discriminative target domain information. Some studies leverage the tracklet information [14] to mitigate the absence of labels. Mekhazni *et al.* [14] propose to align the source and target domain dissimilarity spaces using Maximum Mean Discrepancy. These UDA approaches rely on an abundance of data in the target dataset which may not be feasible for VI-ReID. Moreover, UDA approaches that rely on tracklets which work for single modality Person ReID would fail for VI-ReID. This is because the tracklets would only provide images for a single modality at a time, and obtaining tracklet information for the same identity in both modalities would require knowledge about the identity, thus requiring supervision, which is not feasible.

### 2.3 Label Efficient VI-ReID

Liang *et al.* [10] made one of the earliest attempts to study VI-ReID as an unsupervised learning problem. They propose to first learn the intra-modality feature representations and then use heterogeneous learning to learn shared discriminative feature representations by distilling knowledge from intra-modality pseudo-labels. Subsequent research efforts, such as [20, 23, 24], further explore Unsupervised Learning based VI-ReID (USL-VI-ReID), leveraging various strategies

including camera-level information [23], graph-based structures [20], and cross-modal correspondence mining [24]. Recently, a few studies [1, 4] have delved into UDA for VI-ReID (UDA-VI-ReID), aiming to adapt pre-trained models from a source VI-ReID dataset to a target VI-ReID dataset. Unsupervised methods are a step closer to practical VI-ReID systems as they mitigate the issue of laborious annotations. However, these methods often rely on datasets like SYSU-MM01 and RegDB, which offer ample cross-modal correspondences. Without sufficient cross-modality correspondences, as is the practical scenario, these methods would show deteriorating performance.

Another line of label-efficient approaches study VI-ReID from a semi supervised perspective [16, 18, 22]. These approaches leverage a combination of labelled visible data and unlabelled infrared data for model training. For instance, Wang *et al.* [18] propose Optimal-Transport Label Assignment (OTLA) to tackle this by leveraging an optimal-transport strategy to assign pseudo labels from visible to infrared modality. Shi *et al.* [16] extend upon OTLA to study Semi-Supervised VI-ReID by labelling a portion of the large-scale training data. In contrast to prior studies, we address data and label scarcity by annotating a small set of images from both modalities in the target dataset and eliminating the need for additional unlabelled data, thus achieving data and label efficiency while reducing computational overhead.

### 3 Methodology

In this section, we first formulate the SDA problem for VI-ReID and briefly introduce Maximum Mean Discrepancy (MMD), and then move on to the proposed HD-MMD loss and SDA-VI-ReID framework.

#### 3.1 SDA Problem Formulation

Let  $D_s^v$  and  $D_s^i$  denote the annotated source domain visible and infrared datasets, respectively, where  $D_s^v = \{(v_s^m, y_s^m)\}_{m=1}^{N_s^v}$  and  $D_s^i = \{(i_s^m, y_s^m)\}_{m=1}^{N_s^i}$ . Similarly, let  $V_t$  and  $I_t$  denote the annotated target domain visible and infrared datasets, respectively, where  $D_t^v = \{(v_t^m, y_t^m)\}_{m=1}^{N_t^v}$  and  $D_t^i = \{(i_t^m, y_t^m)\}_{m=1}^{N_t^i}$ . Let  $n_s$  and  $n_t$  denote the number of identities and  $N_s$  and  $N_t$  denote the total number of images in the training set of source and target dataset, respectively. We have,  $n_t \ll n_s$  and  $N_t \ll N_s$ .

Let  $f_\phi$  be a generic embedding network trained on the source domain  $\mathcal{S} = \{D_s^v, D_s^i\}$ . The goal of Supervised Domain Adaptation is to adapt  $f_\phi$  to the target domain  $\mathcal{T} = \{D_t^v, D_t^i\}$ . Note that adaptation involves achieving satisfactory retrieval performance in the target domain.

#### 3.2 Maximum Mean Discrepancy

MMD [5] is a measure used to quantify the discrepancy between two probability distributions. In brief, MMD calculates the difference between the empirical

means of two given sets of samples in a Reproducing Kernel Hilbert Space. For simplicity, MMD can be interpreted as taking a weighted average of the difference of moments between the two distributions by transforming the variables using the kernel  $k$ .

MMD between two distributions  $P$  and  $Q$  can be computed as:

$$\begin{aligned} \text{MMD}^2(P, Q) &= \frac{1}{n_P^2} \sum_{i=1}^{n_P} \sum_{j=1}^{n_P} k(x_i, x_j) \\ &\quad - \frac{2}{n_P n_Q} \sum_{i=1}^{n_P} \sum_{j=1}^{n_Q} k(x_i, y_j) + \frac{1}{n_Q^2} \sum_{i=1}^{n_Q} \sum_{j=1}^{n_Q} k(y_i, y_j) \end{aligned} \quad (1)$$

Where  $k$  represents the kernel function,  $x_i$  and  $y_j$  are samples from distributions  $P$  and  $Q$  respectively, and  $n_P$  and  $n_Q$  are the number of samples from each distribution. We choose the gaussian kernel for  $k$ , given by:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2)$$

MMD has been extensively used in closed-set domain adaptation [21] to minimize the distribution shift between a source domain and a target domain having overlapping label spaces. But MMD can not be directly applied for VI-ReID as the label space of the source and target domains are different (since the identities are different). To circumvent this shortcoming of ambiguous alignment, we introduce HD-MMD.

### 3.3 Hetero-Dissimilarity based Maximum Mean Discrepancy

Since domain adaptation for VI-ReID is an open set problem, generally with no overlap of source and target label spaces, instead of aligning the source and target domains directly, we align the dissimilarity spaces of the two domains. The dissimilarity space [14] of a feature space is a vector space formed by the pairwise dissimilarities of the features. Specifically, we design HD-MMD loss, which aims to align the dissimilarity spaces of the source and target domains, both of which have heterogeneous data.

Directly using D-MMD [14] doesn't help with the retrieval performance (Table 3) because the source and target batches contain heterogeneous data. As shown in Fig. 3 (c), for a robust embedding model, the clusters of the same identity are close but separated between modalities. The modality discrepancy affects the distribution of pairwise distances, leading to reduced performance when D-MMD is directly applied to source and target batches. To address this, we propose aligning the dissimilarity spaces of each modality independently. The bridge between modalities is established using the supervised loss functions of the baseline. The motivation for this approach is formed by the following: 1) D-MMD demonstrates remarkable efficacy in single-modality UDA for Person

Re-identification [14]. 2) Current supervised methods designed for VI-ReID effectively mitigate the modality gap [3, 11, 26]. By incorporating HD-MMD alongside these supervised losses, we effectively form well-structured homogeneous clusters and concurrently diminish the heterogeneous modality gap.

Let  $f_\phi^m$  denote the embedding network for the modality  $m = \{v, i\}$ . For the modality  $m$  and domain  $\delta = \{s, t\}$ , the intra-class (within class) dissimilarity for identity  $i$  between images  $x_i^u, x_i^w \in D_\delta^m$  is given in Eq. 3.  $u$  and  $w$  are 2 different indices for the images of identity  $i$ . Note that we choose the  $L_2$  distance as the dissimilarity measure between 2 vectors to work with a Euclidean Dissimilarity vector space.

$$d_W^{m,\delta}(x_i^u, x_i^w) = \|f_\phi^m(x_i^u) - f_\phi^m(x_i^w)\|_2, \quad u \neq w \quad (3)$$

Similarly, the inter-class (between class) dissimilarity between identities  $i$  &  $j$  for images  $x_i^u, x_j^w \in D_\delta^m$  is given by:

$$d_B^{m,\delta}(x_i^u, x_j^w) = \|f_\phi^m(x_i^u) - f_\phi^m(x_j^w)\|_2, \quad i \neq j \quad (4)$$

We define the MMD loss for the within-class ( $d_W^{m,\delta}$ ) and between-class ( $d_B^{m,\delta}$ ) dissimilarity space of modality  $m$  as:

$$\mathcal{L}_{W,MMD}^m = MMD(d_W^{m,s}, d_W^{m,t}) \quad (5)$$

$$\mathcal{L}_{B,MMD}^m = MMD(d_B^{m,s}, d_B^{m,t}) \quad (6)$$

The final HD-MMD loss is formed by summing up the pair-wise distance losses of both the modalities.

$$\mathcal{L}_{HD-MMD} = \mathcal{L}_{W,MMD}^v + \mathcal{L}_{B,MMD}^v + \mathcal{L}_{W,MMD}^i + \mathcal{L}_{B,MMD}^i \quad (7)$$

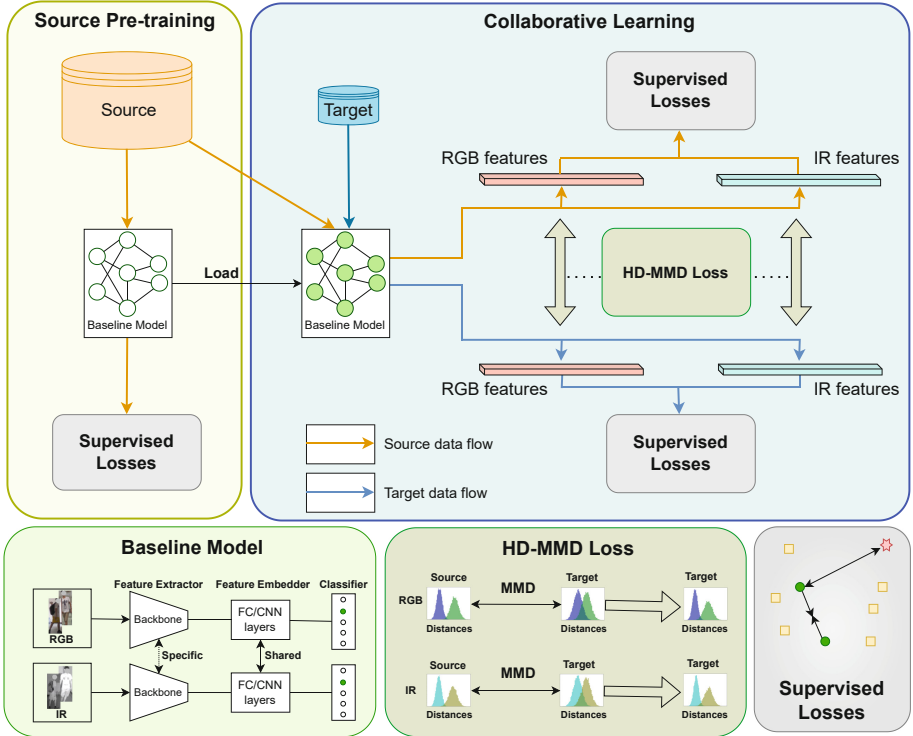
### 3.4 SDA-VI-ReID Framework

SDA-VI-ReID framework can be easily integrated with any existing baseline VI-ReID method. All supervised learning based baselines use the identity loss  $L_{id}$ , along with some metric learning based losses (eg. Triplet Loss or some variant) [11, 26]. For convenience, we will collectively term them as  $L_{sup}^\delta$ , where  $\delta = \{s, t\}$  denotes the domain (source/target).

Herein, we present the two-stage training methodology for SDA-VI-ReID and the training objective employed for Collaborative Learning. Fig. 2 depicts the two-stage framework of the approach, incorporating a generic baseline model for the embedding network alongside the HD-MMD loss.

**Stage 1: Source Pre-Training** In this stage, we train the model  $f_\phi$  using the supervised losses of the baseline with the suggested hyperparameters. Note that the classifier will have  $n_s$  number of output logits in this stage.

**Stage 2: Collaborative Learning** We load the trained model  $f_\phi$  and drop the classifier from stage 1. We append a new classifier having  $n_s + n_t$  number of



**Fig. 2.** The complete SDA-VI-ReID framework. In the first stage, the model is pre-trained on the source dataset. Subsequently, in the second stage, source and target domains collaboratively train the model and HD-MMD is used to bridge the domain shift for RGB and IR modalities. Note that the baseline model and supervised losses are baseline method specific. For the experiments, we choose the baseline as AGW. Typically, ResNet-50 is used as the backbone of the baseline model.

logits and we train the model  $f_\phi$  using the overall training objective  $\mathcal{L}_{tot}$  given by:

$$\mathcal{L}_{tot} = \mathcal{L}_{sup}^t + \lambda_s \cdot \mathcal{L}_{sup}^s + \lambda_h \cdot \mathcal{L}_{HD-MMD} \quad (8)$$

## 4 Experiments

### 4.1 Datasets and Metrics

We evaluate the proposed method on two public VI-ReID datasets: SYSU-MM01 [19] and RegDB [15]. Following the settings in [19], we employ Cumulative Matching Characteristic (CMC) and mean Average Precision (mAP) as evaluation criteria. Additional information about the datasets is provided in the supplementary materials.

## 4.2 Implementation Details

For comparison, we use the standard AGW [26] as our baseline for comparison with state-of-the-art (SOTA). We keep all the hyperparameters as suggested in [26]. For the second stage, we empirically choose  $\lambda_s = 0.25$  and  $\lambda_h = 1$ . To ensure fair comparison, images were augmented using the standard random flipping, random cropping and random erasing [30] strategies, consistent with other SOTA methods [18, 22]. In each batch, we select 4 identities and 4 images from each modality per identity, for both the domains, making a batch of  $2 * 4 * (4 + 4) = 64$  images. Throughout the section, " $x\%$  of dataset" refers to using  $x\%$  of identities from the training set. For example, 20% of RegDB means utilising RGB and IR images from 41 IDs. Further details can be found in supplementary materials.

## 4.3 Results and Analysis

We compare SDA-VI-ReID (ours) with 3 categories of approaches: Fully Supervised methods (*SVI-ReID*), Unsupervised methods (*USVI-ReID*) and Label-Efficient methods (*LEVI-ReID*). SVI-ReID methods use 100% of the labelled training data, whereas our approach uses a small fraction which is mentioned in brackets alongside "SDA". USVI-ReID includes both UDA-VI-ReID and USL-VI-ReID approaches. LEVI-ReID includes Semi-Supervised Learning based approaches. Note that unlike our method, other LEVI-ReID methods use a combination of labelled and unlabelled training data. Since we are the first to study VI-ReID in the data-efficient setting, we mainly compare our method with the closely related LEVI-ReID based approaches.

Overall, our approach outperforms all label-efficient methods on RegDB, as evidenced by Table 1. Using a stronger baseline, we beat unsupervised methods as well while requiring significantly less data and computational resources. It's important to note that the performance on the SYSU-MM01 dataset is constrained by the RegDB dataset's inadequacy as a large-scale source domain. Notably, even 20% of the SYSU-MM01 dataset surpasses the entire RegDB training dataset. Moreover, SYSU-MM01 employs a more extensive camera setup for both modalities, resulting in more robust learned representations compared to RegDB. This highlights the suitability of SYSU-MM01 as a source domain dataset, contrasting with the inadequacy of RegDB, as confirmed by our findings. This limitation adversely affects results on SYSU-MM01, as evident from Table 2. We believe that employing another large-scale VI-ReID dataset as the source domain could substantially enhance results on SYSU-MM01.

**Comparison with LEVI-ReID methods:** Label-efficient (semi-supervised) methods use labelled visible data along with unlabelled infrared data. Overall, our findings reveal that we outperform existing LEVI-ReID based methods on the RegDB dataset (target). This stems from the utilization of the richly labelled and extensive SYSU-MM01 dataset as the source domain while using RegDB as the target domain dataset. We observe that we beat OTLA by using only 20% of the target training data. Moreover, by using just 40% annotations, we consistently outperform all existing semi-supervised methods and even a few SVI-ReID methods too. Conversely, in the alternative scenario, RegDB proves

**Table 1.** Comparison on RegDB (Target) using SYSU-MM01 as Source. † Indicates results are taken from [18]. ‡ Denotes the results without camera information.

Settings			RegDB			
			Visible2Thermal		Thermal2Visible	
Type	Method	Venue	Rank-1	mAP	Rank-1	mAP
SVI-ReID	JSIA-ReID [25]	AAAI'20	48.5	49.3	48.1	48.9
	AGW [26]	TPAMI'21	70.1	66.4	70.5	65.9
	FMCNet [28]	CVPR'22	89.1	84.4	88.4	83.9
	PartMix [8]	CVPR'23	85.7	82.3	84.9	82.5
	SGIEL [3]	CVPR'23	92.2	86.6	91.1	85.2
USVI-ReID	D-MMD† [14]	ECCV'20	2.2	3.7	2.0	3.6
	GLT† [29]	CVPR'21	2.9	4.5	6.3	7.6
	H2H [10]	TIP'21	14.1	12.3	13.9	12.7
	OTLA [18]	ECCV'22	32.9	29.7	32.1	28.6
	ADCA [24]	MM'22	67.2	64.1	68.5	63.8
	PGM [20]	CVPR'23	69.5	65.4	69.9	65.2
	GUR‡ [23]	ICCV'23	73.9	70.2	75.0	69.9
LEVI-ReID	OTLA [18]	ECCV'22	49.9	41.8	49.6	42.8
	TAA [22]	TIP'23	62.2	56.0	63.8	56.5
	DPIS [16]	ICCV'23	62.3	53.2	61.5	52.7
<b>Ours</b>	SDA(20%)	-	51.1	46.9	47.3	44.7
	SDA(40%)	-	<b>71.7</b>	<b>68.1</b>	<b>69.3</b>	<b>66.0</b>

to be an inadequate source domain dataset due to its lack of scale and limited variability within the images. We would like to highlight that, unlike other LEVI-ReID methods, we do not require any additional unlabelled training data from the target domain.

**Comparison with USVI-ReID methods:** We see that we beat all unsupervised methods except GUR by using AGW and 40% data on RegDB. Note that D-MMD and GLT are designed for single modality person re-identification and the results are taken from [18]. We would like to highlight that our approach requires very limited computational resources compared to the unsupervised methods. For analysis of time and memory requirements, please refer to the supplementary materials.

**Comparison with SVI-ReID methods:** We beat AGW while using only 40% of the RegDB data. Remarkably, from Fig. 5 (b), it is evident that we achieve 90.2% Rank-1 accuracy by using 100% of the RegDB with AGW as the baseline, thereby, surpassing all of the SVI-ReID approaches except SGIEL. Note that using SDA with recent baselines would lead to even better results, as evident from Table 4. However, there is still ample room for improvement.

#### 4.4 Ablation Study

In this section, we investigate the influence of the baseline model (AGW), the training stages within the framework, and the HD-MMD objective on the retrieval performance for the RegDB dataset. Additionally, we compare the impact of the standard D-MMD versus our proposed HD-MMD loss, with the same weight for both the losses ( $= 1$ ). The results are summarized in Table 3. Since our approach relies on a large-scale source domain dataset, our experiments are performed using SYSU-MM01 as the source and RegDB as the target dataset.

**Table 2.** Comparison on SYSU-MM01 (Target) using RegDB as Source. <sup>†</sup> Indicates results are taken from [18]. <sup>‡</sup> Denotes the results without camera information.

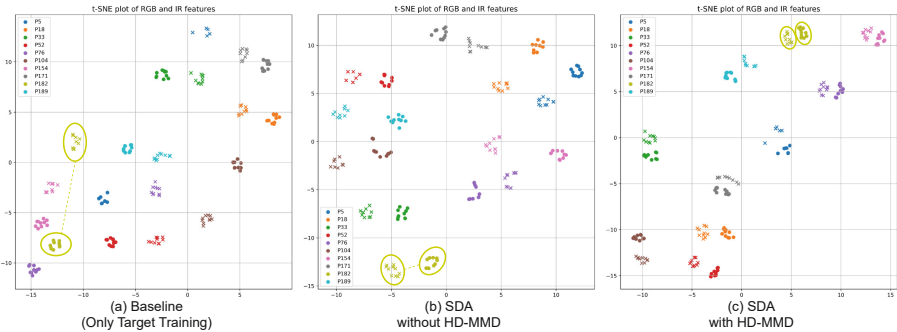
Settings			SYSU-MM01			
			All Search		Indoor Search	
Type	Method	Venue	Rank-1	mAP	Rank-1	mAP
SVI-ReID	JSIA-ReID [25]	AAAI’20	38.1	36.9	43.8	52.9
	AGW [26]	TRAMI’21	47.5	47.7	54.2	63.0
	FMCNet [28]	CVPR’22	66.3	62.5	68.2	74.1
	PartMix [8]	CVPR’23	77.8	74.6	81.5	84.4
	SGIEL [3]	CVPR’23	77.1	72.3	82.1	82.9
USVI-ReID	D-MMD <sup>†</sup> [14]	ECCV’20	12.5	10.4	19.0	15.4
	GLT <sup>†</sup> [29]	CVPR’21	7.7	9.5	12.1	18.0
	H2H [10]	TIP’21	25.5	25.2	-	-
	OTLA [18]	ECCV’22	29.9	27.1	29.8	38.8
	ADCA [24]	MM’22	45.5	42.7	50.6	59.1
	PGM [20]	CVPR’23	57.3	51.8	56.2	62.7
	GUR <sup>‡</sup> [23]	ICCV’23	61.0	57.0	64.2	69.5
LEVI-ReID	OTLA [18]	ECCV’22	48.2	43.9	47.4	56.8
	TAA [22]	TIP’23	48.8	42.3	50.1	56.0
	DPIS [16]	ICCV’23	<b>58.4</b>	<b>55.6</b>	<b>63.0</b>	<b>70.0</b>
<b>Ours</b>	SDA(20%)	-	26.6	26.4	27.4	36.3
	SDA(40%)	-	36.0	36.3	39.2	50.1

Row 1 corresponds to training solely on the target dataset with only 20% of the data. Training the baseline directly on the small target dataset yields poor performance, but as we incorporate components of our proposed approach, retrieval performance improves. Remarkably, employing only Stage-2 of the SDA-VI-ReID framework yields comparable results, indicating proper alignment of the source domain with a sufficient number of epochs, and subsequent alignment of



**Table 3.** Ablation Study on 20% of RegDB (Target) with SYSU-MM01 as Source. S-1 and S-2 refer to Stage 1 and 2 of our SDA-VI-ReID framework. The Baseline is AGW. Row number 1 refers to only target training without using the source domain.

No.	Method					Visible2Thermal		Thermal2Visible	
	Baseline	S-1	S-2	D-MMD	HD-MMD	Rank-1	mAP	Rank-1	mAP
1	✓					29.4	28.7	28.2	27.6
2	✓		✓			39.2	40.6	38.9	39.4
3	✓	✓	✓			40.1	38.6	40.5	39.9
4	✓	✓	✓	✓		44.0	39.6	43.5	41.1
5	✓		✓		✓	49.4	46.3	<b>47.8</b>	43.2
6	✓	✓	✓		✓	<b>51.1</b>	<b>46.9</b>	47.3	<b>44.7</b>



**Fig. 3.** t-SNE plots showing RegDB test features of the same 10 IDs. Baseline refers to direct target training using AGW baseline. Circles: RGB, Crosses: IR features.

the target domain via the HD-MMD loss. This significantly reduces computational overhead, as Stage-1 can be omitted while achieving similar outcomes. Moreover, as evident from rows 4 and 6, our HD-MMD loss surpasses D-MMD, highlighting the suitability of our approach for domain adaptation across heterogeneous domains. Visualization via t-SNE [13] plots in Fig. 3 demonstrates that clusters of the same identity converge closer upon employing SDA and HD-MMD. Additionally, clusters become more compact with HD-MMD, signifying improved learning of the cluster structure from aligned dissimilarity spaces for both modalities.

Further, Table 4 demonstrates the effectiveness of our proposed SDA framework and HD-MMD loss by showcasing significant performance improvements when integrated with several popular methods, including AGW [26], HcTri [11], and MMD-ReID (abbr. MMDR) [6]. Only Target (row 1) refers to training the baseline directly on 20% of RegDB. HcTri and MMDR utilize stronger supervised signals, which can be unstable when directly applied to the small target dataset and lead to feature degradation. However, when paired with our framework, the

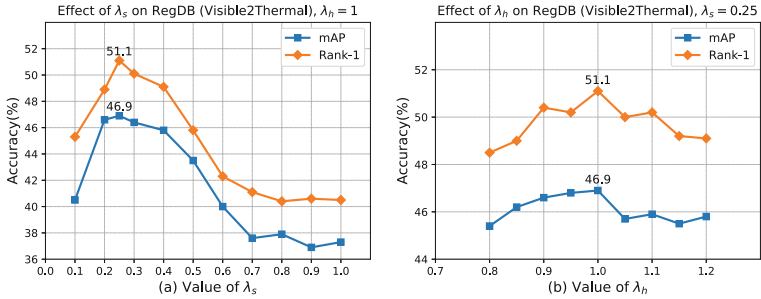
learnt representations are better. This highlights the seamless integration of our framework with recent methods and its ability to enhance their performance.

#### 4.5 Sensitivity Analysis

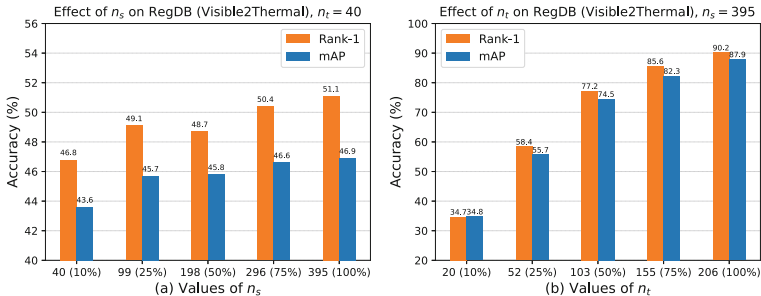
**Effect of  $\lambda_s$  and  $\lambda_h$ :** The influence of hyperparameters  $\lambda_s$  and  $\lambda_h$  on mAP and Rank-1 accuracy is illustrated in Fig. 4. We observe a steady decline in performance as  $\lambda_s$  increases, indicating a shift in the optimization focus towards the source domain at the expense of the target domain. Further, we see that the HD-MMD loss, governed by  $\lambda_h$ , exhibits stability in the range  $[0.8, 1.2]$ , demonstrating the robustness of the proposed loss.

**Table 4.** Improvement in performance of 3 different VI-ReID methods for 20% RegDB (Visible2Thermal) by integrating SDA and HD-MMD. SYSU-MM01 is used as source domain for SDA.

Method	AGW [26]		HcTri [11]		MMDR [6]	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
Only Target	29.4	28.7	18.5	17.7	20.7	19.3
With SDA	40.1	38.6	54.2	50.7	59.2	54.5
SDA+HD-MMD	<b>51.1</b>	<b>46.9</b>	<b>60.1</b>	<b>56.3</b>	<b>63.7</b>	<b>58.9</b>



**Fig. 4.** Effect of  $\lambda_s$  and  $\lambda_h$  on 20% RegDB (Target) with SYSU-MM01 as Source.



**Fig. 5.** Effect of  $n_s$  and  $n_t$  on RegDB (Target) with SYSU-MM01 as Source.

**Effect of number of source IDs  $n_s$  and target IDs  $n_t$  in Stage-2:** Fig. 5 shows the variation of mAP and Rank-1 accuracy with the  $n_s$  and  $n_t$ . As expected, increasing the number of target domain IDs results in an improvement in retrieval performance. Remarkably, we achieve 90.2% Rank-1 accuracy by using all of the RegDB training data, whereas the baseline AGW only achieves 70.1% Rank-1 (Table 1). This demonstrates the versatility of our approach in not only improving retrieval performance for small target domains, but also improving performance of fully supervised approaches on large target datasets by leveraging a large-scale source dataset. The influence of source domain size on target domain accuracy is apparent from Fig. 5 (a), supporting our claim that a large-scale source dataset facilitates learning better representations for the target domain.

## 5 Conclusion

This paper proposes SDA-VI-ReID, a simple two-stage learning framework for data-efficient VI-ReID. In the initial pre-training stage, we leverage a large-scale source domain dataset to train a robust embedding network. Subsequently, in a collaborative learning approach during the second stage, we adapt this network to a data-scarce target domain. We show that by choosing an appropriately large-scale source domain along with a strong baseline, robust representations can be learned for the target domain via the proposed framework. Additionally, we introduce the HD-MMD loss, which aligns heterogeneous source and target domains, effectively leveraging the scarce cross-modal correspondences of the target domain. Furthermore, we demonstrate that the proposed approach seamlessly integrates with existing VI-ReID baselines to improve their performance. This study pioneers the investigation of domain adaptive VI-ReID from a pragmatic supervised standpoint, while also presenting an innovative framework for data and label efficient VI-ReID. Through rigorous experimentation and ablation analysis, we have demonstrated the effectiveness of the proposed methodology. The insights gained from our study provide a solid foundation for future investigations into domain adaptation and data-efficient approaches in VI-ReID.

## References






1. Chen, Q., Quan, Z., Li, Y., Zhai, C., Mozerov, M.G.: An unsupervised domain adaption approach for cross-modality rgb-infrared person re-identification. *IEEE Sensors Journal* (2023)
2. Feng, H., Chen, M., Hu, J., Shen, D., Liu, H., Cai, D.: Complementary pseudo labels for unsupervised domain adaptation on person re-identification. *IEEE Trans. Image Process.* **30**, 2898–2907 (2021)
3. Feng, J., Wu, A., Zheng, W.S.: Shape-erased feature learning for visible-infrared person re-identification. In: *CVPR*. pp. 22752–22761 (2023)

4. Fu, X., Huang, F., Zhou, Y., Ma, H., Xu, X., Zhang, L.: Cross-modal cross-domain dual alignment network for rgb-infrared person re-identification. *IEEE Trans. Circuits Syst. Video Technol.* **32**(10), 6874–6887 (2022)
5. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. *The Journal of Machine Learning Research* **13**(1), 723–773 (2012)
6. Jambigi, C., Rawal, R., Chakraborty, A.: Mmd-reid: A simple but effective solution for visible-thermal person reid. In: *BMVC* (2021)
7. Jiang, K., Zhang, T., Liu, X., Qian, B., Zhang, Y., Wu, F.: Cross-modality transformer for visible-infrared person re-identification. In: *ECCV*. pp. 480–496. Springer (2022)
8. Kim, M., Kim, S., Park, J., Park, S., Sohn, K.: Partmix: Regularization strategy to learn part discovery for visible-infrared person re-identification. In: *CVPR*. pp. 18621–18632 (2023)
9. Lee, G., Lee, S., Kim, D., Shin, Y., Yoon, Y., Ham, B.: Camera-driven representation learning for unsupervised domain adaptive person re-identification. In: *ICCV*. pp. 11453–11462 (2023)
10. Liang, W., Wang, G., Lai, J., Xie, X.: Homogeneous-to-heterogeneous: Unsupervised learning for rgb-infrared person re-identification. *IEEE Trans. Image Process.* **30**, 6392–6407 (2021)
11. Liu, H., Tan, X., Zhou, X.: Parameter sharing exploration and hetero-center triplet loss for visible-thermal person re-identification. *IEEE Trans. Multimedia* **23**, 4414–4425 (2021)
12. Liu, J., Sun, Y., Zhu, F., Pei, H., Yang, Y., Li, W.: Learning memory-augmented unidirectional metrics for cross-modality person re-identification. In: *CVPR*. pp. 19366–19375 (2022)
13. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(86), 2579–2605 (2008)
14. Mekhazni, D., Bhuiyan, A., Ekladios, G., Granger, E.: Unsupervised domain adaptation in the dissimilarity space for person re-identification. In: *ECCV*. pp. 159–174. Springer (2020)
15. Nguyen, D.T., Hong, H.G., Kim, K.W., Park, K.R.: Person recognition system based on a combination of body images from visible light and thermal cameras. *Sensors* **17**(3), 605 (2017)
16. Shi, J., Zhang, Y., Yin, X., Xie, Y., Zhang, Z., Fan, J., Shi, Z., Qu, Y.: Dual pseudo-labels interactive self-training for semi-supervised visible-infrared person re-identification. In: *ICCV*. pp. 11218–11228 (2023)
17. Song, L., Wang, C., Zhang, L., Du, B., Zhang, Q., Huang, C., Wang, X.: Unsupervised domain adaptive re-identification: Theory and practice. *Pattern Recogn.* **102**, 107173 (2020)
18. Wang, J., Zhang, Z., Chen, M., Zhang, Y., Wang, C., Sheng, B., Qu, Y., Xie, Y.: Optimal transport for label-efficient visible-infrared person re-identification. In: *ECCV*. pp. 93–109. Springer (2022)
19. Wu, A., Zheng, W.S., Yu, H.X., Gong, S., Lai, J.: Rgb-infrared cross-modality person re-identification. In: *ICCV*. pp. 5380–5389 (2017)
20. Wu, Z., Ye, M.: Unsupervised visible-infrared person re-identification via progressive graph matching and alternate learning. In: *CVPR*. pp. 9548–9558 (2023)
21. Yan, H., Li, Z., Wang, Q., Li, P., Xu, Y., Zuo, W.: Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation. *IEEE Trans. Multimedia* **22**(9), 2420–2433 (2019)

22. Yang, B., Chen, J., Ma, X., Ye, M.: Translation, association and augmentation: Learning cross-modality re-identification from single-modality annotation. *IEEE Trans. Image Process.* **32**, 5099–5113 (2023)
23. Yang, B., Chen, J., Ye, M.: Towards grand unified representation learning for unsupervised visible-infrared person re-identification. In: *ICCV*. pp. 11069–11079 (2023)
24. Yang, B., Ye, M., Chen, J., Wu, Z.: Augmented dual-contrastive aggregation learning for unsupervised visible-infrared person re-identification. In: *ACM MM*. pp. 2843–2851 (2022)
25. Yang, Y., Zhang, T., Cheng, J., Hou, Z., Tiwari, P., Pandey, H.M., et al.: Cross-modality paired-images generation and augmentation for rgb-infrared person re-identification. *Neural Netw.* **128**, 294–304 (2020)
26. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.: Deep learning for person re-identification: A survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(6), 2872–2893 (2021)
27. Ye, M., Wang, Z., Lan, X., Yuen, P.C.: Visible thermal person re-identification via dual-constrained top-ranking. In: *IJCAI*. vol. 1, p. 2 (2018)
28. Zhang, Q., Lai, C., Liu, J., Huang, N., Han, J.: Fmcnet: Feature-level modality compensation for visible-infrared person re-identification. In: *CVPR*. pp. 7349–7358 (2022)
29. Zheng, K., Liu, W., He, L., Mei, T., Luo, J., Zha, Z.J.: Group-aware label transfer for domain adaptive person re-identification. In: *CVPR*. pp. 5310–5319 (2021)
30. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: *AAAI*. **34**, 13001–13008 (2020)



# Anonymisation for Time-Series Human Activity Data

Tim Hallyburton<sup>1</sup> , Nilah Ravi Nair<sup>2</sup> , Fernando Moya Rueda<sup>3</sup> ,  
René Grzeszick<sup>3</sup> , and Gernot A. Fink<sup>1</sup> 

<sup>1</sup> Department of Computer Science, TU Dortmund University, Dortmund, Germany  
[tim.hallyburton@tu-dortmund.de](mailto:tim.hallyburton@tu-dortmund.de)

<sup>2</sup> Chair of Material Handling and Warehousing, TU Dortmund University,  
Dortmund, Germany  
[nilah.nair@tu-dortmund.de](mailto:nilah.nair@tu-dortmund.de)

<sup>3</sup> MotionMiners GmbH, Dortmund, Germany

**Abstract.** Time-series human activity data obtained from sensor technologies facilitate various applications in industry and daily life, such as activity recognition, motion or fall detection, and health analysis. Recent research has shown that person re-identification and soft-biometric recognition are feasible from these activity recordings, leading to privacy breaches. Consequently, anonymising the subject characteristics found in the sensor recordings while retaining data utility is of interest. Here, we present an anonymisation framework using a conditioned autoencoder-based GAN that allows for three anonymisation strategies for time-series human activity data experimented on two complementary datasets. The framework was visually verified with experiments on motion capture data before being applied to inertial measurement data. This framework reduces re-identification to 0.52% while maintaining data utility for activity recognition tasks. Further, we present a form of anonymisation using identity transfer with the help of deep feature interpolation. The method achieves over 96% successful identity transfer with high data utility.

**Keywords:** Anonymising · Privacy · GAN · DFI · Autoencoder

## 1 Introduction

Privacy refers to the autonomy of the disclosure, usage, and availability of one's personal or otherwise confidential information [21]. Though complex, the concept of privacy is a topic of interest due to the recent developments in artificial intelligence (AI) and the increased possibility of malicious use of data. Consequently, governments have brought forth regulations, such as the General Data Protection Regulation (GDPR) [4] and the AI Act [5], to protect individuals from fraudulence and distress with AI. One method of mitigating the fear of data misuse is anonymizing personal data before saving it on a third-party system. ISO/IEC 25237:2017 [9] defines anonymisation as the process by which

personal data is irreversibly modified such that subject data can no longer be retraced directly or indirectly by the data user alone or in collaboration with any other party. As a result, through anonymization, data can be used for various analyses and applications while retaining the data provider’s privacy.

Human motion data has immense potential in fall detection, activity recognition, and health analysis. Recent studies have identified that motion data can be used for person re-identification or soft-biometrics recognition (e.g., age, gender, and height) using time-series sensor data [16]. Consequently, there is a need to anonymize the subject characteristics from the time-series data while maintaining the data utility or minimizing the identifiable features. Previous works like [22] have attempted differential privacy, augmentation, and synthetic data to remove or complement original data. However, very little work exists for anonymisation in the domain of human activity data. As a result, this paper brings forth anonymisation strategies for sensor-based human activity data from a known subject re-identification network on two human activity datasets.

This work considers a situation where a dataset created for an application, such as human activity recognition (HAR), is repurposed to identify the subjects performing the activities [16], thus compromising the subjects’ privacy. Given the scenario, this work explores the possibility of generating synthetic data of the subjects performing the activities while removing the subject-specific information from the recordings. Specifically, can generative networks be used to develop methods for anonymising sensitive user data while preserving its similarity to the original data and thus ensuring privacy protection?

For this task, this paper proposes three strategies of anonymisation, where the generative model attempts to create an identity space where mutual information can be removed or interpolated while retaining motion information that is close to reality for an application such as HAR. The first strategy considers a situation where the generative model attempts to remove information by removing the generic subject information present in the re-identification model. The second strategy assumes knowledge of the subject performing the activity and thus directs the generative model to remove that subject-specific information. The final strategy assumes the knowledge of a subject whose identity was compromised and can be the target identity of the generator.

The paper is organised as follows: Section 2 presents recent work in anonymisation and generative networks. Section 3 explains the networks that facilitate the three strategies. Section 4 discusses the results obtained. Finally, the conclusions and future work are presented in Section 5.

## 2 Related work

Human activity data drive innovative technology such as human-machine interaction, virtual and augmented reality, and simulation environments [23]. Though not intended for the use of person and soft-biometric identification [16] and enhancing video-based subject tracking [7], time-series data unwittingly provide person-specific information present in the motion recordings of the person

to machine learning methods. For example, deep neural networks (DNNs) can extract subject-specific information from stagnant and locomotive activities [16].

Differential privacy (DP) suggests that small alterations to the recorded time-series data do not significantly affect the statistical properties of the dataset [3]. Consequently, there has been high interest in evaluating DP, especially on time series data. For example, [22] applied DP with the addition of controlled noise in time-series data of electric footprints of smart homes for preserving privacy. However, the data utility was found to be insufficient. This work emphasizes the computational challenges of adding noise to preserve privacy while maintaining the statistical structure of the data.

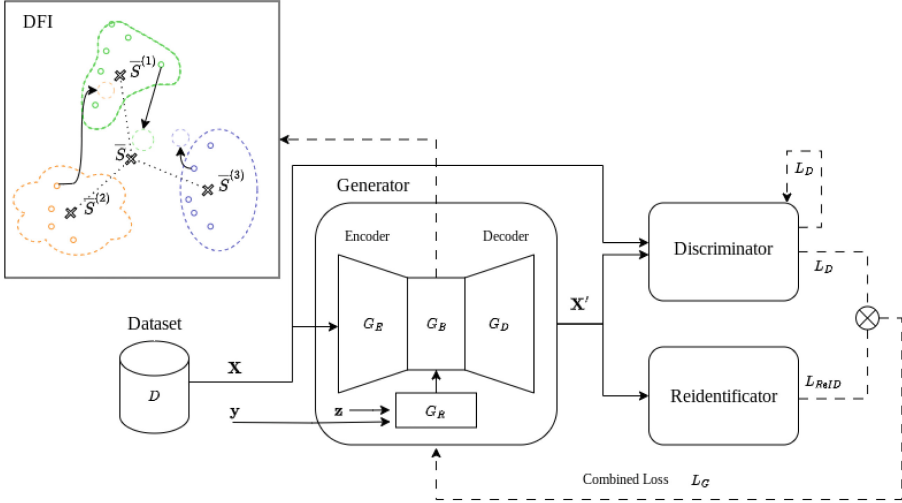
One of the few works that explore inertial measurement unit (IMU) data privacy preservation is [12]. The authors present two approaches utilising autoencoders (AEs) for the privacy preservation of smartphone data. First, the recorded data is categorised based on the target application requirement as *required*, *neutral*, and *sensitive* segments. The sensitive time-series recordings are either replaced with a Replacement autoencoder (RAE) or anonymised using an Anonymising autoencoder (AAE). On the one hand, the RAE is trained to categorize the time-series data into categories and then replaces the *sensitive* timeframes with randomly chosen *neutral* timeframes. On the other hand, AAE attempts to minimize privacy loss while maximising data utility. The authors calculate the privacy loss as the mutual information of user-specific data in the anonymised time series and the user’s identity. Several use cases on different datasets were successfully implemented and evaluated with both networks and a combination in which data is first masked with an RAE and then further obscured with an AAE. Their results show that re-identification accuracy can drop from 96.2% down to 7.0% while keeping activity recognition at high levels [12].

Generative Adversarial Networks (GANs) are not primarily used for IMU time series data generation and augmentation. The research focus on GANs has preliminarily been on art, entertainment, medical imaging, drug discovery, and financial modelling [10]. One instance of GANs being used to generate IMU data is TheraGAN. TheraGAN is a conditional GAN trained to generate realistic IMU signals to elevate imbalances in the activity classes, leading to more robust classifiers for therapeutic application. The synthesised data can replace individual channels of IMU recordings without impairing subsequent HAR models [13]. Consequently, exploring the possibility of using GANs for privacy-preserving synthetic data generation is interesting. For instance, [8] and [2] use GANs for full-body de-identification or anonymisation of subjects in image data. Deep neural networks such as convolutional neural networks (CNNs) map complex features into a Euclidean subspace, where the features can be disentangled, and linearised [20], as the authors of [1]’s hypothesised. Deep feature interpolation (DFI) uses linear interpolation within local subspaces to achieve precise and controlled modifications of attributes, for example, in face images, adding or removal of beard, glasses or skin properties [20], while keeping the face identity intact [18]. To the best of our knowledge, this method has not been reciprocated



for sensor-based human motion data. As a result, this work focuses on anonymisation strategies implemented with AE-based GAN architectures and subject feature transfer using DFI.

### 3 Anonymisation using Generative Networks



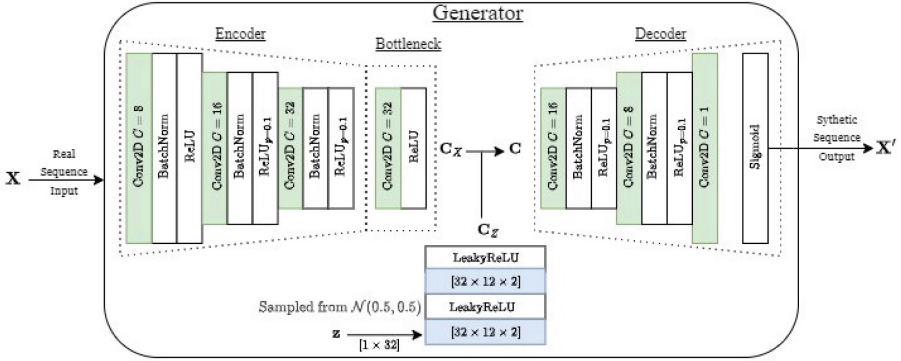
**Fig. 1.** Architectural overview of the proposed method. Real data  $X$  from data set  $D$  is anonymised in the generator network. The autoencoder structure allows for controlled deep feature interpolation (DFI) at the bottleneck, making re-identification of the generator’s output  $X'$  less likely. The generator is trained on the dual objective as given by the discriminator and identifier networks combined loss  $L_G$ .

We propose a framework for anonymising multichannel time-series sequence recordings of humans when performing activities using generative networks. For a triple  $(X, a, y)$  with  $X$  a sequence of sensor recordings,  $a$  the activity, and  $y$  the identity label of the subject, this framework seeks to generate  $(X', a', y')$ , so that  $y' \neq y$  and  $a' = a$ . Figure 1 shows the overall framework. This anonymisation framework consists of an adversarial architecture with three main components: an **autoencoder generator** that generates synthetic data, a **discriminator** that attempts to differentiate between synthetic and authentic input, and a **re-identification network** that recognises the subject performing the activity of the input-segmented recording.<sup>1</sup>

The following are the key points considered in the design of the anonymisation strategies. Firstly, the method assumes that the re-identification network

<sup>1</sup> The code and parameters of the networks are available on [GitHub](#).

that an attacker can use is known. Secondly, two constraints are placed on the generative networks to ensure data utility and a broader range of real-world applications: the anonymisation process is conducted on segmented recordings, and the anonymisation strategy should be oblivious to the activities in the segments. Consequently, a pre-trained re-identification network is used to help train the anonymisation networks. Further, the generated anonymised recordings are tested on a pre-trained HAR network that was not included in the training process.



**Fig. 2.** The Autoencoder generator network has (De-)Convolutional layers in the network with filter size  $[4 \times 4]$  (green layers) to implement the en- and decoding of the code representation  $\mathbf{C}$ . The code representation is manipulated by adding the latent features  $\mathbf{C}_z$  as learned in the dense branch of the network (Blue represents fully-connected layers). Encoder, Bottleneck and Decoder are pre-trained, with the encoder having fixed parameters during the GAN training phase.

The autoencoder (AE) generator architecture structurally uses a deep, convolutional AE. Deep convolutional networks are efficient at solving HAR and re-identification tasks [14, 16]. Besides, convolutions are efficient at feature extraction that can facilitate DFI [18]. The autoencoder structure has the additional benefit of simplifying the reverse mapping problem as the autoencoder learns to decode autonomously [20]. Figure 2 shows the AE generator architecture. The encoder of the AE consists of three convolutional layers, with batch normalisation and ReLU activations. The bottleneck layer has a single convolutional layer followed by ReLU activations. The decoder consists of three de-convolutional layers, with the first two layers containing batch normalisation and ReLU activation. A sample recording  $\mathbf{X}$  is passed through the encoder, yielding the deep representation  $\mathbf{C}_x$ . Passing this deep representation to the decoder yields the reconstructed data sequence  $\mathbf{X}'$  of the same shape as  $\mathbf{X}$ . The AE of the generator is pre-trained to guarantee a viable reconstruction process and to establish a baseline concerning HAR and *ReID* performance.

While training the GAN, a latent vector  $\mathbf{z}$  is added to the encoder deep representation  $\mathbf{C}_x$ , after feature extraction through two dense layers with leaky

ReLU activations. These dense layers, thus, provide the possibility to model and embed  $\mathbf{C}_z$  that can manipulate the encoder deep representation  $\mathbf{C}_x$  based on the combined loss of the GAN. Both,  $\mathbf{C}_x$  and  $\mathbf{C}_z$  have the same dimensions; their weighted sum is denoted as  $\mathbf{C}$ . The addition of controlled noise to the embedding can be considered a variation of differential privacy (DP) as in [6], where the authors argued that the sampling process for VAE described a DP variant. In the case of subject-based conditioning of the generator, the latent vector  $\mathbf{z}$  is concatenated with the one-hot vector of the subject label  $\mathbf{y}$ . The new vector is passed through the dense layers for extracting the feature  $\mathbf{C}_z$ . Thus, instead of directly adding noise to the bottleneck, this process provides freedom to add randomness, as well as, condition the bottleneck layer.

The re-identifier *ReID* serves as a metric to assess the effectiveness of the generator’s anonymisation. The *ReID* architecture assigns a block of four convolutional layers to each limb recording. These convolutional layers operate in parallel and are then fused by flattening and concatenation. A multi-layer perceptron (MLP) with softmax activation yields the identity prediction for each subject. The discriminator, denoted as  $D$ , is realised using a CNN architecture with three convolutional layers, with each layer followed by batch normalization and ReLU activation. The extracted features are then provided to a fully connected layer. This output is obtained through a Sigmoid activation function in the final layer of the CNN.

### 3.1 Anonymisation Strategies

The anonymisation framework addresses three strategies, giving a general solution for anonymisation. The first strategy removes the generic information of the subjects present in the dataset without specifically focusing on the subject performing the activity in the given segmented recording, called *Anon<sub>AG</sub>*. The second strategy imposes a condition on the generators’ learning based on the subject’s identity performing the activities in the given segmented recording, called *Anon<sub>AcG</sub>*. The *Anon<sub>AcG</sub>* has the base structure of *Anon<sub>AG</sub>* but is conditioned on subject identity. However, the conditional value is provided to the generator and not the discriminator. Inspired by DFI, the third strategy interpolates subject representation from the pretrained AE generators’ embeddings of the segmented recordings. Further, the generative model is trained to transfer a target subject’s identity onto the generated synthetic segment. Thus, the DFI-based GAN architecture, *Anon<sub>DFI</sub>*, performs anonymisation through subject feature transfer.

*Anon<sub>AG</sub>* and *Anon<sub>AcG</sub>* focus on reducing subject re-identification while generating synthetic sequences with high data utility. Thus, a combination of loss functions achieves effective anonymisation and synthesis, where an inverted binary cross-entropy (BCE) loss for  $D$ ,  $BCE(D[\mathbf{X}^l])$ , is multiplied by the cross-entropy loss of *ReID* for the source subject  $CE(ReID[\mathbf{X}]^{(y)})$ . However, when the *ReID* prediction for the actual identity approaches zero, indicating successful concealment of the identity, the entire loss term collapses to a negligible value. Consequently, the discriminator’s influence is nullified, resulting in the

network generating random, unidentifiable noise, which is undesirable. Thus, a linear clipping of the *ReID* loss to a minimum value ensures that the *ReID* output is low. A loss value  $l$ ,  $l \mapsto m \cdot l + b$  reduces the *ReID* loss but is fully differentiable, Equation (1).

$$L_G(\mathbf{X}', y) = BCE(D[\mathbf{X}']) \cdot (m \cdot CE(ReID[\mathbf{X}]^{(y)}) + b), \text{ for } m, b \in \mathbb{R}^+, m + b = 1, \quad (1)$$

The combination loss affects the learning of the feature  $\mathbf{C}$ . For instance,  $\mathbf{C}_x$  learns features most sensitive to re-identification. Applying an inverted *ReID*-loss to the dense layer that maps the latent vector  $\mathbf{z}$  to  $\mathbf{C}_z$  enforces a heavy distortion of identifiable features while maintaining data utility provided, the losses of *ReID* and  $D$  are combined. The difference between the methods *Anon<sub>AG</sub>* and *Anon<sub>AcG</sub>* generators is that, in *Anon<sub>AcG</sub>*, the subject identity is provided to the generator as an encoding concatenated to latent vector  $\mathbf{z}$ . This process increases uncertainty as the identity labels are fed exclusively to the generator, not the discriminator. This deliberate deviation from the standard implementation provides a context for the generator to learn anonymisation. However, this modification does not affect the loss Equation (1).

The training of the *Anon<sub>DFI</sub>* is complementary to the training of *Anon<sub>AG</sub>* and *Anon<sub>AcG</sub>* strategies. In *Anon<sub>DFI</sub>*, the focus is to reduce/remove the identity of the subject performing the activity in the given segmented recording and to replace a target subject’s identity in its place. Consequently, the first step in this direction is to remove the mean subject-specific information. The subject-specific feature encoding can be obtained from the bottleneck of the autoencoder. Thus, for a given sequence  $X$  of a source subject  $y \in Y$  with feature representation  $\mathbf{C}_x$ , the mean subject-specific information across all  $N$  sequence representations is described as  $\bar{\mathbf{S}}^{(y)} = \frac{1}{N} \sum_{i=0}^N \mathbf{C}_i^{(y)}$ .

Adapting the concept of DFI, a linear transformation exists in the feature space that allows the shift of identity to a feature representation resembling another subject. However, this transformation cannot be naively applied to the bottleneck encoding for the reason that the feature space consists of subject identity as well as activity features that are entangled. Given the activity-agnostic training, the feature entanglement cannot be easily resolved. The dataset imbalance amplifies the feature entanglement problem. Consequently, *Anon<sub>DFI</sub>* minimises the mean subject-specific features from the given sequence and adds the target subject features learned through the GAN training, as presented in Equation (2). To facilitate the target subject feature learning process, the target identity embedding is concatenated to the latent vector input  $\mathbf{z}$ , which is further encoded as  $\mathbf{C}_z(y')$  through the deep layers discussed previously.

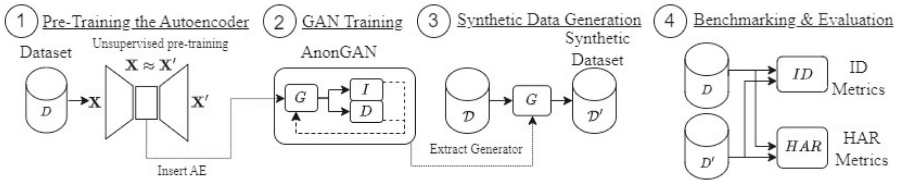
$$\mathbf{C} = (\mathbf{C}_x - \bar{\mathbf{S}}_y) + \mathbf{C}_z(y') \quad (2)$$

This difference in training requirement implies that the loss function Equation (1) must be updated. Consequently, the linear scaling of the *ReID*’s contribution to the total loss was dropped, thus alleviating the concern about vanishing

gradients associated with the function, Equation (3). The new loss function  $L_G^{DFI}$  is minimal if the discriminator perceives the synthetic samples  $\mathbf{X}'$  as authentic and simultaneously, the re-identification  $G_I$  assigns them with high probability to the target subject  $y'$ .

$$L_G^{DFI}(\mathbf{X}', y') = BCE(D[\mathbf{X}']) \cdot CE(ReID[\mathbf{X}'], y') \quad (3)$$

### 3.2 Training



**Fig. 3.** Overview of the experimental process. Step (1) describes the unsupervised training of an autoencoder. (2) This autoencoder is inserted into the GAN network structure and the GAN training commences. After the training, the generator is used for synthetic data generation in step (3). Step (4) then compares the synthetic and original datasets using HAR and ID networks.

We follow a multi-phase training process outlined in Figure 3. Initially, the autoencoder architecture of the generator is trained independently. The primary objective of this phase is to identify an optimal embedding for the activity data, which will be consistently used in the later stages. Furthermore, this step establishes a preliminary quality benchmark for the data reconstruction. The parameters of the autoencoder are then fixed and transferred into the generator. Next, the pre-trained identification,  $ReID$ , with fixed parameters and the untrained discriminator  $D$  are integrated into the network.  $ReID$  is fixed to represent an adversary network attempting re-identification. In contrast, the discriminator gets trained parallel to the generator to ensure consistent similarity between real and synthetic data for subsequent applications.

## 4 Anonymisation Results

The anonymisation method presented here combines the three established strategies, deep feature interpolation, differential privacy, and a GAN structure, two of which are implemented directly in the generator. This approach is evaluated on two publicly available benchmark datasets of inertial measurement recordings of human movements.

## 4.1 Datasets

Two datasets were used for experimentation, namely the Logistic Activity Recognition Challenge (LARA) dataset (version 2) [17] and the MotionSense [11] dataset. LARA consists of both motion capture (MoCap)  $LARA_{MoCap}$  and inertial measurement unit (IMU) data  $LARA_{IMU}$ , whereas MotionSense (MS) consists of only IMU data.  $LARA_{IMU}$  has a total of five on-body devices (OBDS) with three sensors each, whereas MS has one OBD with three sensors. The  $LARA_{MoCap}$  is sampled at  $200Hz$ , while  $LARA_{IMU}$  and MS are sampled at  $100Hz$  and  $50Hz$ , respectively. The datasets consist of varied sets of activities and subjects. The MS data set has 24 subjects performing six different activities of daily living, while  $LARA_{MoCap}$  and  $LARA_{IMU}$  have 16 and 7 subjects, respectively, performing seven classes of logistics activities. The MoCap data from LARA is used to visually verify the methods used in the experiment. The work, however, is focused on the anonymisation of IMU data.

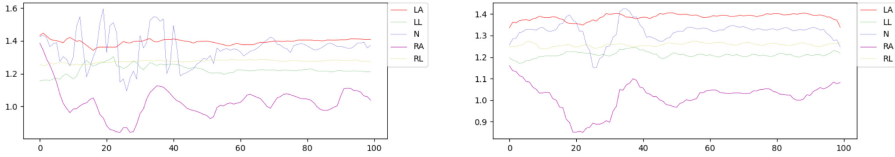
## 4.2 Benchmarked networks

The anonymisation method, see Figure 3, depends on two architectures controlling the quality of the anonymisation process, such that the anonymised sequences will fool the ReID while the HAR prediction accuracy is maintained. Thus, the sequence will contain information allowing HAR but not ReID. These two control-point architectures, ReID and HAR are benchmarked architectures proposed in [16, 15].

Two variations of the ReID can be found based on the number of channels in the datasets. For datasets with high channel density, the ReID network, as detailed in [16], is utilised and referred to as  $ReID_L$ . For example,  $ReID_L$  [16] is used for the  $LARA_{MoCap}$  and  $LARA_{IMU}$ , further denoted as  $ReID_L^{MC}$  and  $ReID_L^{IMU}$ , respectively. In the case of low channel density, a single block of four convolutional layers performs the feature extraction, referred to as  $ReID_M$ . The pre-trained ReID networks assess the possibility of re-identification from the synthesised recordings.  $ReID_M$  is used mainly with the MS dataset, referred to as  $ReID_M^{MS}$ .

Similarly, pre-trained HAR networks for  $LARA_{MoCap}$ ,  $LARA_{IMU}$ , and MS data were used to quantify the utility of the generated data for HAR applications. Based on the work of [19], tCNN-IMU networks were used for  $LARA_{MoCap}$  and  $LARA_{IMU}$ , referred to as  $CNNIMU_{MC}$  and  $CNNIMU_{IMU}$ , respectively. For the MS, a  $CNNIMU_{MC}$  with one branch is used for HAR, as mentioned in [19].

**Autoencoder** Two variations of AEs with different features in the bottleneck layer were experimented on. The first variant has 32 features, and the second variant has 64 features. Minor padding is applied to the input and output layers to make the chosen filter size fit the data shape. For the (de-)convolutional layers, a fixed filter size of  $[4 \times 4]$  with a stride of 2 was effective during a hyperparameter search. The AE is trained unsupervised, and the network generalises while being agnostic about input recording activities. Mean Squared Error (MSE) loss was



**Fig. 4.** Visual inspection of the reconstruction quality of  $AE_{MbientLab}$ . The real sequence  $\mathbf{X}$  (left) from the MbientLab dataset is fed to the respective autoencoder network. The reconstruction yielded by  $AE_{MbientLab}$  (right) retains the general trajectories of the time-series but misses out on details, showing a quantisation effect on temporal neighbourhood.

employed. The networks with 64 features at the bottleneck performed best at low validation loss and visual inspection as presented in Figure 4.

Table 1 presents the baseline performance obtained for the HAR and ReID networks on the real dataset and the AE-generated data for the datasets of interest. A learning rate of 0.0001, batch size 50 and epoch 10 for LARa dataset ReID network and a learning rate of 0.001, batch size 50 and epoch 100 for MS ReID network was found to be effective.

**Table 1.** Baseline ReID and HAR Metrics for  $LARa_{MoCap}$ ,  $LARa_{IMU}$  and *Motion-Sense*.

Dataset	Real Data				$AE_{Synth}$ Data			
	ReID		HAR		ReID		HAR	
	Acc%	wF1%	Acc%	wF1%	Acc%	wF1%	Acc%	wF1%
$LARa_{MoCap}$	98.97	98.97	76.54	76.19	99.81	99.81	97.19	97.18
$LARa_{IMU}$	94.61	94.57	80.28	79.81	78.50	77.55	64.60	63.98
<i>MS</i>	78.23	78.08	95.81	95.75	9.70	5.85	55.80	53.73

Generally, a drop in network performance can be seen on the data synthesised by the AE. Interestingly, network trained on synthetic  $LARa_{MoCap}$  performs better than the real data. One could attribute this performance difference to the high channel density MoCap data being optimised with the encoding of the AE. However, in comparison, loss of information from the low channel density IMU data is evident.

### 4.3 $Anon_{AG}$

Table 2 presents the baseline results of the HAR and ReID networks using the synthetic data obtained from the autoencoder-based GAN,  $Anon_{AG}$ , trained on each dataset.  $Anon_{AG}$  trained on  $LARa_{IMU}$  achieved optimal results at the 20-epoch mark. In contrast,  $Anon_{AG}$  trained on  $LARa_{MoCap}$  achieved stable training after 5 epochs. The experiments run best at a small learning rate of  $1 \times 10^{-5}$

**Table 2.** Benchmark for synthetic data generated by the respective models compared to real data. Values correspond to averages across 5 runs (mean  $\pm$  std. deviation). High HAR metrics indicate good utility preservation, while low ID metrics show successful anonymisation.

Network	Anon <sub>AG</sub>				Real Data			
	HAR		ReID		HAR		ReID	
	Acc%( $\uparrow$ )	wF1%( $\uparrow$ )	Acc%( $\downarrow$ )	wF1%( $\downarrow$ )	Acc%	wF1%	Acc%	wF1%
Anon <sup>MoCap</sup> <sub>AG</sub>	37.45 $\pm$ 0.22	26.57 $\pm$ 0.16	<b>6.52 <math>\pm</math> 0.10</b>	<b>1.19 <math>\pm</math> 0.03</b>	76.54	76.19	98.97	98.97
Anon <sup>IMU</sup> <sub>AG</sub>	<b>44.77 <math>\pm</math> 0.28</b>	<b>35.23 <math>\pm</math> 0.24</b>	13.91 $\pm$ 0.08	4.52 $\pm$ 0.13	80.28	79.81	94.61	94.57
Anon <sup>MS</sup> <sub>AG</sub>	36.36 $\pm$ 0.12	33.47 $\pm$ 0.12	7.12 $\pm$ 0.07	2.30 $\pm$ 0.05	95.81	95.75	78.23	78.08

for both the generator and discriminator. Anon<sub>AG</sub> was efficient in anonymising LARa<sub>IMU</sub> dataset. Unaltered data on the ReID network with 95% accuracy reduced to 4.5% on the synthesised data. Some of the data utility was lost in this process as  $wF_1$  score of the activity recognition on the original data at 80% dropped significantly to 35% for the Anon<sub>AG</sub> on LARa<sub>IMU</sub>. A similar drop in performance can be seen in the LARa<sub>MoCap</sub> performance. Specifically, low ReID was achieved at 6.52%. These results show that the generator learned to modify privacy-sensitive features but compromised the integrity of the time series, leading to lower HAR accuracy than the benchmark values. This outcome can be associated with the absence of guidance for the generator regarding the identity information to be concealed. The Anon<sub>AG</sub> performs generic anonymisation by applying modifications indiscriminately. Thus negatively affecting data utility.

#### 4.4 Anon<sub>AcG</sub>

The second strategy provides the subject information to the generator. The identity information provided is a conditioning that allows Anon<sub>AcG</sub> to learn about the subject-specific characteristics it must mask to facilitate anonymisation. The hyperparameter search on this architecture shows that the generator’s training process was stable at a low learning rate. Overfitting was addressed by employing a EXPONENTIALLR learning rate scheduler that dynamically adjusts the learning rates for both the generator and the discriminator after each epoch.

**Table 3.** Benchmark for synthetic data generated by the respective models compared to real data. Values correspond to averages across 5 runs (mean  $\pm$  std. deviation). High HAR metrics indicate good utility preservation, while low ID metrics show successful anonymisation.

Network	Anon <sub>AcG</sub>				Real Data			
	HAR		ReID		HAR		ReID	
	Accuracy ( $\uparrow$ )	wF1 ( $\uparrow$ )	Accuracy ( $\downarrow$ )	wF1 ( $\downarrow$ )	Acc%	wF1%	Acc%	wF1%
Anon <sup>MoCap</sup> <sub>AcG</sub>	36.42 $\pm$ 0.21	25.38 $\pm$ 0.15	6.46 $\pm$ 0.09	1.19 $\pm$ 0.02	76.54	76.19	98.97	98.97
Anon <sup>IMU</sup> <sub>AcG</sub>	<b>63.14 <math>\pm</math> 0.17</b>	<b>60.08 <math>\pm</math> 0.18</b>	<b>0.76 <math>\pm</math> 0.01</b>	<b>0.52 <math>\pm</math> 0.01</b>	80.28	79.81	94.61	94.57
Anon <sup>MS</sup> <sub>AcG</sub>	37.94 $\pm$ 0.13	35.53 $\pm$ 0.11	5.26 $\pm$ 0.07	1.97 $\pm$ 0.05	95.81	95.75	78.23	78.08



The respective HAR and ReID metrics were established to compare  $Anon_{AcG}$  with benchmark values. Table 3 presents the average  $Acc$  and  $wF1$  over five training-test sets.  $Anon_{AcG}$  achieves an identification accuracy of 0.76% while maintaining high data utility for  $LARa_{IMU}$ . Furthermore,  $Anon_{AcG}$  generates samples matching the predicted activity of the original data in 63.14% of the cases, whereas samples generated by  $Anon_{AG}$  yielded 44.77% accuracy for this benchmark. Similar results can be seen with the MS dataset, too. However,  $LARa_{Mocap}$  performance decreases, specifically for ReID. The visible data utility improvement can be attributed to including the identity labels in the network’s input.

#### 4.5 $Anon_{DFI}$

$Anon_{DFI}$  deviates from the previously explained philosophy of anonymisation as it attempts a controlled identity transfer for a given sequence, being aware of the source and desired target identity. As discussed in Section 3, the bottleneck of the AE generator encodes features that can be shifted in a desired direction following the concept of DFI.

**Table 4.** Benchmark of HAR and ReID metrics per target subject an example on  $LARa_{IMU}$ . Higher metrics correspond to better preservation of activity information and successful anonymisation through identity transfer to the target subject.

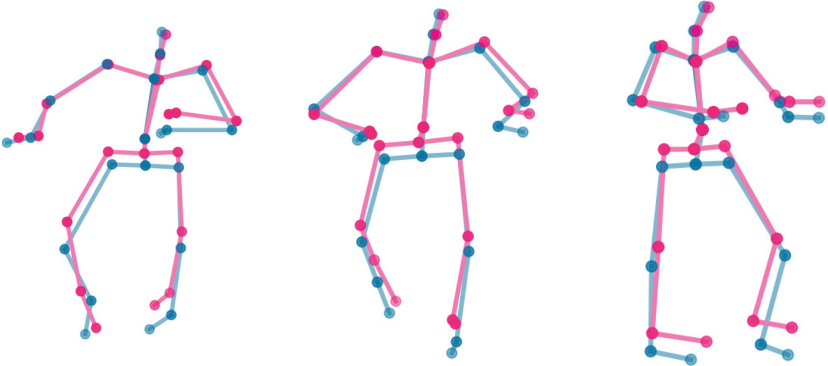
Target Subject	HAR Metrics		ReID Metrics	
	$wF_1$ ( $\uparrow$ )	Accuracy ( $\uparrow$ )	$wF_1$ ( $\uparrow$ )	Accuracy ( $\uparrow$ )
$S_0$	$58.00 \pm 0.28$	$59.68 \pm 0.21$	$99.63 \pm 0.01$	$99.26 \pm 0.03$
$S_1$	$56.62 \pm 0.15$	$59.80 \pm 0.12$	$99.88 \pm 0.00$	$99.77 \pm 0.00$
$S_2$	$59.03 \pm 0.33$	$62.25 \pm 0.27$	$99.71 \pm 0.01$	$99.42 \pm 0.02$
$S_3$	$54.60 \pm 0.27$	$58.38 \pm 0.26$	$99.67 \pm 0.01$	$99.34 \pm 0.03$
$S_4$	$51.08 \pm 0.20$	$53.53 \pm 0.16$	$99.72 \pm 0.01$	$99.43 \pm 0.01$
$S_5$	$52.87 \pm 0.31$	$57.40 \pm 0.34$	$99.71 \pm 0.00$	$99.42 \pm 0.00$
$S_6$	$59.69 \pm 0.31$	$62.08 \pm 0.38$	$99.26 \pm 0.04$	$98.53 \pm 0.07$
$S_7$	$57.94 \pm 0.18$	$61.51 \pm 0.17$	$99.68 \pm 0.01$	$99.35 \pm 0.01$

A preliminary test of subject transfer applied solely on the AE provided encouraging results, as presented in Table 4. For instance, DFI-based subject transfer on  $LARa_{IMU}$  achieves good data utility preservation, comparable to the results of  $Anon_{AcG}$ . Figure 5 presents a comparison between the original skeleton of Subject 08 in ■ interpolated to subject 15 in ■ from  $LARa_{Mocap}$ . However, the anonymization is much weaker, with an average Re-ID accuracy of 30%, compared to the previously achieved 0.7%. Compared to the AE baseline, the results demonstrate improved data utility preservation with HAR accuracies of 80% while concurrently reducing ReID scores by half. Thus, motivating the

method discussed in Section 3 to train the GAN, referred to as  $Anon_{DFI}$ , to achieve enhanced data utility and anonymisation through subject transfer.

**Table 5.** Benchmark for synthetic data generated by the respective models compared to real data. Values correspond to averages across 5 runs (mean  $\pm$  std. deviation). High HAR metrics indicate good utility preservation, while low ID metrics show successful anonymisation.

Network	$Anon_{DFI}$				Real Data			
	HAR		ReID		HAR		ReID	
	Accuracy ( $\uparrow$ )	$wF_1$ ( $\uparrow$ )	Accuracy ( $\downarrow$ )	$wF_1$ ( $\downarrow$ )	Acc%	wF1%	Acc%	wF1%
$Anon_{DFI}^{MoCap}$	37.45 $\pm$ 0.22	26.57 $\pm$ 0.16	<b>6.52 <math>\pm</math> 0.10</b>	<b>1.19 <math>\pm</math> 0.03</b>	76.54	76.19	98.97	98.97
$Anon_{DFI}^{IMU}$	<b>44.77 <math>\pm</math> 0.28</b>	<b>35.23 <math>\pm</math> 0.24</b>	13.91 $\pm$ 0.08	4.52 $\pm$ 0.13	80.28	79.81	94.61	94.57
$Anon_{DFI}^{MS}$	39.37 $\pm$ 0.12	35.48 $\pm$ 0.11	5.94 $\pm$ 0.04	2.20 $\pm$ 0.03	95.81	95.75	78.23	78.08



**Fig. 5.** Comparison between the original skeleton in ■ with the generated one using the  $Anon_{DFI}$  from source Subject 08 to Subject 15 in ■ from the  $LARaMoCap$ .

The  $Anon_{DFI}$  training process significantly increased the GAN’s stability and positively influenced the quality of the generated samples.  $Anon_{DFI}^{IMU}$  achieved stability at low epochs of 4 and 7 epochs, respectively. As this method focuses on targeting the entire dataset to a target subject, the maximal identity metric scores for the target subject indicate the best privacy preservation. This adjustment of the evaluation strategy ensures that the results are not misleading due to the imbalanced support across subjects in the dataset. The network reliably transfers over 96% of all sequences to any target subject while maintaining high data utility. We observe a notable variance in the HAR metrics, depending on which subject is chosen as the target. A possible reason for this observation can be found in the composition of the dataset used, as slightly over-represented subjects are preferred to be targets.

## 5 Conclusion

The objective of this work was to explore and develop a privacy-preserving framework that maintains data utility for IMU data in the context of HAR using generative networks. This framework consists of an adversarial architecture conditioned on a discriminator and an identification network to transform input sequences such that future re-identification is impossible. Notice that the framework is not conditioned to the activities performed by the subjects in the recordings. The framework considers three different anonymisation strategies: without subject information, with subject information and anonymisation through subject feature transfer. These strategies cover different use-case scenarios of anonymisation.

Three GAN architectures in alliance with the presented strategies were implemented. *Anon<sub>AcG</sub>* yielded the best results by lowering the re-identification  $wF_1$  score from 80.28% to 0.52%, while maintaining HAR scores above 60%. Furthermore, two distinct approaches to subject feature transfer have been introduced and experimentally verified, conditioning the GAN for interpolating deep representation of subjects.

The findings demonstrate the effectiveness of a GAN-based network architecture in reducing re-identification risks associated with IMU data and open the following topics for further exploration. Firstly, the effect of the anonymised samples generated by GANs on the training of HAR models embedded in end-user devices must be investigated. Anonymisation directly on end-user devices allows for fully preserving privacy. However, the performance of this method can be contrasted with federated learning. Second, it would be insightful to investigate whether the developed targeted DFI architecture can effectively contribute to balancing datasets, specifically by enhancing the representation of subjects, thus, addressing the dataset bias of HAR models.

**Acknowledgment.** The work in this publication was supported by the German Federal Ministry of Education and Research (BMBF) in the context of the project “LAMARR Institute for Machine Learning and Artificial Intelligence” (Funding Code: LAMARR24B).

## References






1. Bengio, Y., Mesnil, G., Dauphin, Y., Rifai, S.: Better Mixing via Deep Representations (Jul 2012), [arXiv:1207.4404](https://arxiv.org/abs/1207.4404) [cs]
2. Brkic, K., Sikiric, I., Hrkac, T., Kalafatic, Z.: I know that person: Generative full body and face de-identification of people in images. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1319–1328 (2017). <https://doi.org/10.1109/CVPRW.2017.173>
3. Dwork, C.: Differential Privacy: A Survey of Results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) Theory and Applications of Models of Computation, vol. 4978, pp. 1–19. Springer, Berlin Heidelberg, Berlin, Heidelberg (2008)

4. European, C.: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016), published: Official Journal of the European Union
5. European Parliament: Artificial Intelligence Act: Deal on comprehensive rules for trustworthy AI (2023)
6. Groß, B., Wunder, G.: Differentially Private Synthetic Data Generation via Lipschitz-Regularised Variational Autoencoders (Jul 2023), [arXiv:2304.11336](https://arxiv.org/abs/2304.11336) [cs]
7. Henschel, R., Von Marcard, T., Rosenhahn, B.: Simultaneous Identification and Tracking of Multiple People Using Video and IMUs. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 780–789. IEEE, Long Beach, CA, USA (Jun 2019)
8. Hukkelås, H., Lindseth, F.: Deepprivacy2: Towards realistic full-body anonymization. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1329–1338 (2023)
9. ISO: ISO/IEC 25237:2017 Health Informatics. Pseudonymization. pub-ISO, 1 edn
10. Li, Z., Xia, B., Zhang, J., Wang, C., Li, B.: A Comprehensive Survey on Data-Efficient GANs in Image Generation (Oct 2022), [arXiv:2204.08329](https://arxiv.org/abs/2204.08329) [cs]
11. Malekzadeh, M., Clegg, R.G., Cavallaro, A., Haddadi, H.: Mobile sensor data anonymization. In: Proceedings of the International Conference on Internet of Things Design and Implementation. pp. 49–58. IoTDI '19, ACM, New York, NY, USA (2019)
12. Malekzadeh, M., Clegg, R.G., Cavallaro, A., Haddadi, H.: Privacy and Utility Preserving Sensor-Data Transformations (Nov 2019), [arXiv:1911.05996](https://arxiv.org/abs/1911.05996) [cs, eess, stat]
13. Mohammadzadeh, M., Ghadami, A., Taheri, A., Behzadipour, S.: cGAN-Based High Dimensional IMU Sensor Data Generation for Therapeutic Activities (Feb 2023), [arXiv:2302.07998](https://arxiv.org/abs/2302.07998) [cs]
14. Moya Rueda, F.: Transfer Learning for Multi-Channel Time-Series Human Activity Recognition. PhD Thesis, Technische Universität Dortmund (Sep 2023)
15. Moya Rueda, F., Grzeszick, R., Fink, G., Feldhorst, S., Ten Hoppel, M.: Convolutional Neural Networks for Human Activity Recognition Using Body-Worn Sensors. *Informatics* **5**(2), 26 (May 2018)
16. Nair, N.R., Moya Rueda, F., Reining, C., Fink, G.A.: Multi-channel time-series person and soft-biometric identification. In: International Conference on Pattern Recognition. pp. 256–272. Springer (2022)
17. Niemann, F., Reining, C., Moya Rueda, F., Bas, H., Altermann, E., Nair, N.R., Steffens, J.A., Fink, G.A., ten Hoppel, M.: Logistic Activity Recognition Challenge (LARA Version 02) – A Motion Capture and Inertial Measurement Dataset (Feb 2022)
18. Palyam, R.K.: Deep Feature Interpolation for Image Content Changes. Master's thesis, Technische Universität Dortmund, Dortmund (2018)
19. Rueda, F.M., Fink, G.A.: From Human Pose to On-Body Devices for Human-Activity Recognition. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 10066–10073 (2021)
20. Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep Feature Interpolation for Image Content Changes (Jun 2017), [arXiv:1611.05507](https://arxiv.org/abs/1611.05507) [cs]
21. Wallace, K.A.: Anonymity. *Ethics Inf. Technol.* **1**(1), 21–31 (1999)

22. Wang, H., Wu, C.: Privacy Preservation for Time Series Data in the Electricity Sector. *IEEE Transactions on Smart Grid* **14**(4), 3136–3149 (2023)
23. Wang, M.: A Comprehensive Survey on Human Activity Recognition Using Sensing Technology. *Highlights in Science, Engineering and Technology* **9**, 376–389 (Sep 2022)



# Representation Biases in Time-Series Human Activity Recognition with Small Sample Sizes

Nilah Ravi Nair<sup>1</sup>, Lena Schmid<sup>2</sup>, Christopher Reining<sup>1</sup>,  
Fernando Moya Rueda<sup>5</sup>, Markus Pauly<sup>2,4</sup>, and Gernot A. Fink<sup>3</sup>

<sup>1</sup> Chair of Material Handling and Warehousing, TU Dortmund University, Dortmund, Germany

[nilah.nair@tu-dortmund.de](mailto:nilah.nair@tu-dortmund.de)

<sup>2</sup> Department of Statistics, TU Dortmund University, Dortmund, Germany

[lana.schmid@tu-dortmund.de](mailto:lana.schmid@tu-dortmund.de)

<sup>3</sup> Department of Computer Science, TU Dortmund University, Dortmund, Germany

<sup>4</sup> Research Center Trustworthy Data Science and Security,  
UA Ruhr, Dortmund, Germany

<sup>5</sup> MotionMiners GmbH, Dortmund, Germany

**Abstract.** Neural networks trained on human motion data have various industrial and daily living applications, such as activity recognition, gesture recognition, and gait-based biometrics. These neural network models are often trained on industrial or research datasets designed for a specific application with a narrow subject pool. Given that subject re-identification and soft-biometric, such as age, gender, and height, identification is feasible using neural networks trained on human activity data, the influence of these characteristics on HAR models cannot be ignored. Biased datasets can halt neural networks from generalizing to unseen subjects. However, the biases found in activity data are not explicit. As a result, this paper focuses on representation biases caused by the training data subject characteristics in multi-channel time-series human activity data obtained from sensor technologies. We provide a statistical approach to evaluate the biases in existing datasets, a method to account for biases, and a perspective on subject selection criteria for future human activity datasets. The study is a step towards fair and trustworthy artificial intelligence by attempting to quantify the subject bias in multi-channel time-series HAR data.

**Keywords:** Bias · Human Activity Recognition · Fair AI · Trustworthy AI · Dataset

---

N. R. Nair and L. Schmid—These authors contributed equally to this work.

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_3](https://doi.org/10.1007/978-3-031-78354-8_3).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15315, pp. 33–48, 2025.  
[https://doi.org/10.1007/978-3-031-78354-8\\_3](https://doi.org/10.1007/978-3-031-78354-8_3)

# 1 Introduction

Human activity recognition (HAR) involves recognizing an individual’s physical activities from multi-channel time-series (MCTS) sensor recordings. HAR research is relevant for human technology interaction, mobile, and ubiquitous computing applications for industries and daily living. In many cases, applications use neural network-based classifiers trained on MCTS datasets designed for a specific use case. However, these classifiers’ robustness is determined by the quality of the dataset used [2], as neural networks inherit biases from the datasets [22]. For example, varying sensor placements [5], a shift of the domain [11], inconsistent labels [19], and class imbalance [16], present in the datasets introduce so-called dynamic inductive biases to the classifier. Similarly, subject characteristics in the dataset influence the activity classifier. The authors in [22] name the under-representation of a part of the population that an application targets and the subsequent failure to generalize as representation bias. In accordance with this, this work refers to the biases caused or influenced by subject characteristics represented in the dataset as representation bias.

Person re-identification and soft-biometrics such as age, gender and height identification are feasible with time-series human activity recordings [15, 25]. These works emphasize the influence of an individual’s characteristics in the time-series data. Researchers have attempted cross-validation, personalization of HAR networks, augmentation and synthetic data generation to provide generalized HAR models [4, 9]. However, these attempts do not evaluate or acknowledge representation biases of the datasets. Generalized HAR models or Universal HAR models are defined to be capable of generalizing to motion patterns of any subject [4]. However, achieving such a model is restricted by the availability of datasets that vary in the subject’s physical characteristics and documentation. Creating such datasets is time-consuming due to the efforts for sensor set-up, data recording and cleaning, and labeling [18, 21]. For example, for the LARA dataset [16], annotation alone took 85 min per 2 min of recorded data [19], or 90 min per 1 min for HAR datasets in industry as reported by MotionMiners GmbH. Furthermore, subject selection criteria followed by dataset creators in the dataset creation process are based on the availability of actors or volunteers.

To our knowledge, no approach or metric for time-series human activity data biases is available. Consequently, this work develops an approach to account for representation biases in a dataset, evaluates the representation biases learned by HAR models and thus, provides a subject selection criteria [18], as a form of representation bias mitigation strategy starting from the source – the dataset. Thus, this work aims to be a first step towards ensuring fair and trustworthy models for MCTS HAR applications. The contribution aims to answer the following questions:

*RQ1*: Do the physical characteristics of humans influence activity recognition performance?

*RQ2*: What physical characteristics should be considered when selecting subjects to create a robust classifier?

*RQ3*: Can we develop a metric for representation bias in activity recognition classifiers?

The remainder of this contribution is structured as follows. Section 2 presents works on the motion behavior of humans resulting from their physical characteristics and connection to identity and activity recognition. Section 3 elaborates on an approach to bias evaluation and explains the experimental design to quantify the influence of subject characteristics on HAR performance. Section 4 presents the quantitative results of the experiments using different datasets, and Section 5 presents the answers to each research question. Finally, Section 6 discusses the main contributions and concludes with further work and an outlook.

## 2 Related Work

Datasets are prone to biases. According to [7], eight biases generally found in datasets are social, measurement, representation, label, algorithmic, evaluation, deployment and feedback bias; such datasets further bias data-driven machine learning (ML) methods. Biased ML models can lead to unfair results in sensitive applications such as deep face recognition, loan and credit, and product suggestion applications [3,7]. Consequently, evaluating and mitigating the biases are vital for generalizing ML models. Representation bias, in particular, is associated with the dataset creation process. Thus, to ensure fair and trustworthy ML models, creating a balanced dataset is of interest [10]. The authors created a balanced face dataset that included age, gender, and ethnic aspects. The model facilitated an accurate classification model with the help of the public image dataset with equal representation of each characteristic. In a similar effort, [28] proposed a metric called the calibrated detection rate (recall) of a demographic characteristic for face detection. Furthermore, the authors evaluated various face detection bias mitigation strategies. Similar research in computer vision motivated authors of [27] to create a tool called REVISE, which facilitates the detection of potential biases in a visual dataset for the object, person and geography-based analysis.

Previous research on biases in HAR focuses on dynamic inductive biases, such as the type of sensors, sensor positions, segment size, and pre-processing [8]. For example, [29] uses bias and noise correction formulas for sensor data pre-processing. However, bias caused by the subjects selected for HAR datasets is unexplored [18]. Gait activity-based person re-identification suggests that each individual’s motion behavior is unique and can be referred to as a biometric [1]. Nevertheless, physical characteristics such as height, weight, and handedness can influence the performance of various activities [15,20].

The impact of the representation bias is visible when accounting for the generalization capability of the models. For instance, [12] segregated the HAR models into three categories: personal, impersonal and hybrid. Personal models are trained and tested on the same subject’s activities, while impersonal or universal models use training data from users not present in the test set. Finally, the hybrid model combines the personal and impersonal models. The evaluation



of the models shows that the universal model performs the worst at 76% accuracy. In comparison, personal and hybrid models perform better at 98% and 95% classification accuracy, respectively. However, personal and hybrid models may not be feasible for practical applications, e.g. in industrial settings with frequent staff changes. Thus, evaluating the physical or soft-biometric characteristics of the individual is of interest to create a robust impersonal model of the HAR classifier. The authors in [6] weighted the training data by considering the similarity between subjects of the training and test data. In addition to the similarity of physical attributes, similar signal patterns were evaluated. The authors considered a Euclidean distance between the feature vectors of two subjects based on age, weight and height and visualized a multi-dimensional scaling over physical characteristics. They experimented with their method on the UNIMIB-SHAR, Mobiact and MotionSense datasets and showed that their approach improves the classification accuracy.

Though the above-discussed literature suggests the impact of subjects' individuality on HAR accuracy and introduces a concept to improve classification accuracy with the help of personalization of the model, the issue of generalized HAR models is yet to be tackled. Consequently, understanding the subject characteristics that induce bias, a dataset creation methodology, and a metric to account for the biases are necessary for dataset creators as a subject selection criterion [18] to mitigate the resulting bias.

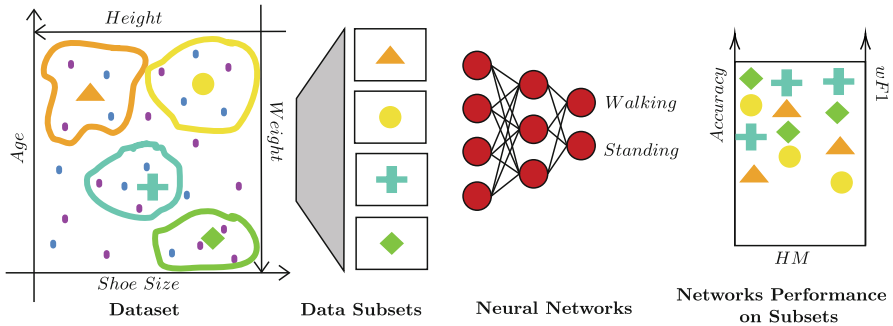
### 3 Statistical Analysis of Representation Bias in Human Activity Recognition

The influence of an individual's physical characteristics is blended into human motion. As a result, isolating the features of the human motion associated with the physical characteristics from the recordings is improbable. Sensor biases, sensor placement, or the idiosyncrasies of the individual's motion cannot be isolated from human movement recorded using on-body devices (OBDs), such as inertial measurement units (IMUs). Consequently, statistical evaluation of the impact of the subject's physical and soft-biometric characteristics is desirable. In this section, we elaborate on the hypotheses, the evaluation strategy, the networks, and the datasets of interest.

#### 3.1 Formulation of Hypotheses

Multi-channel time-series datasets such as Motion Capture (MoCap) systems and OBDs record the human body's movements. The physical characteristics, as well as the soft-biometrics of an individual, influence the motion. For example, an old subject may walk slower than a young subject. However, it is unclear if an OBD placed at the wrist of the subject is influenced by the subject's handedness alone or if the height, gender, and age attributes contribute to biases in HAR datasets. In particular, is such a recording influenced by the subject characteristic or human representation available in the dataset?

Figure 1 outlines the proposed approach. A statistical concept known as heterogeneity measure (HM) is utilized to curate the training set for a neural network. HM quantifies the diversity or non-uniformity of qualities within a dataset, providing insight into the range and distribution of qualities present [17]. This measure helps understand the spectrum of physical and soft-biometric characteristics among subjects in HAR datasets. Thus, an HM-curated training set includes different physical or soft-biometric heterogeneity levels. Here, the hypothesis is that when maintaining the size of the training data across all heterogeneity-level experiments, classifier performance on unseen test sets increases with an increase in heterogeneity of the subject’s physical characteristics in the training data.



**Fig. 1.** A curated subset of the training dataset is selected based on a heterogeneity measure for training neural networks. We hypothesize that classifier performance variation can be identified based on the heterogeneity measure.

### 3.2 The Representation Bias - An Evaluation Strategy

We desire an evaluation strategy that considers the diversity of the subjects’ characteristics in the training set. Thus, we propose a quality measure that depends on the different number of characteristic levels in the dataset. For example, assume two characteristics,  $A$  and  $B$ , each with levels ranging from 1 to  $a$  and 1 to  $b$ , respectively; in this work,  $A$  could represent age, divided into the levels ‘young’ and ‘old’, while  $B$  could represent gender, ‘male’ and ‘female’. Thus, characteristics  $A$  and  $B$  have two levels ( $a = b = 2$ ). This gives us  $a \cdot b (= 4)$  different levels of characteristics. These levels imply  $a \cdot b$  different potential heterogeneity groups ranging from being a completely homogeneous training set where all subjects have the same characteristic level (here referred to as group 1) to a completely heterogeneous training set where all subjects have varied characteristic levels (here, group  $ab$  or 4). Therefore, heterogeneity is gauged based on the various levels in the training sets of the same size, with not all combinations necessarily feasible (depending on the dataset). Within a heterogeneity

group lying between completely homogenous and completely heterogenous, further division into subgroups depending on the characteristic level combinations is feasible, as elaborated in Section 3.4.

For a fair comparison of the heterogeneity group, the amount of data present in the training sets of each group should be approximately equal. As a result, the training sample size for the activity classifier depends on subgroup size; for smaller datasets, the training set size equals  $ab$  subjects, while for larger datasets, a multiple of  $ab$  is used to ensure coverage of all heterogeneity groups. However, it is to be noted that the size must be set before splitting the data into heterogeneity groups such that all the heterogeneity groups have approximately equal numbers of training data. Subjects not included in the training set are reserved for testing the classifier. To mitigate selection bias, subjects are randomly selected for the training set. This allows a comprehensive exploration of the dataset’s diversity, with  $N$  distinct experiments conducted for each heterogeneity group. Experiments are conducted with the maximum number of training sets possible for the respective dataset in cases where  $N$  different training sets were not feasible.

Categorization of physical characteristics, such as age, which are continuous, is necessary to simplify the analysis while preserving important data characteristics. This process facilitates pattern identification and ensures a sufficient sample size for robust statistical analysis. Thus, this approach can be generalized to more characteristics, e.g., a third characteristic  $C$  with levels  $1, \dots, c$  gives  $abc$  heterogeneity groups. Again, heterogeneity is measured from group 1 to group  $abc$ . As this approach becomes cumbersome with large characteristic levels, selecting 4-8 different groups is suggested.

### 3.3 Datasets

The dataset selection criteria for this work were the availability of varied subjects, documentation of subjects’ physical characteristics, the public availability of the dataset, varied activities within the dataset and previous use of the dataset in HAR research. Table 1 presents the chosen datasets for this work. MobiAct [26], Motionsense [13], and Sisfall [24] have one OBD, which typically consists of an accelerometer and gyroscope (IMU). In addition, a MoCap dataset, LARaMoCap [16], is included in the experiment. The three IMU datasets are recordings of activities of daily living such as walking, jogging, and sitting. The

**Table 1.** Datasets for experimentation and their features.

Dataset	Sampling Rate (Hz)	No: Subject	No: Activities	Sensor Placement	Characteristics Available
<i>MobiAct</i> [26]	20	58	9	Trouser Pocket	Age, Gender, Weight, Height
<i>Motionsense</i> [13]	50	24	6	Trouser pocket	Age, Gender, Weight, Height
<i>Sisfall</i> [24]	200	38	15	Waist	Age, Gender, Weight, Height
<i>LARaMoCap</i> [16]	200	16	7	Body joints	Age, Gender, Weight, Height, Handedness

**Table 2.** Statistical summary of the physical characteristics. Weight is measured in kilograms (kg), height in centimeters (cm), and gender is denoted as F for females and M for males.

Dataset	Age					Height					Weight					Gender	
	Min.	1st Qu.	Med.	3rd Qu.	Max.	Min.	1st Qu.	Med.	3rd Qu.	Max.	Min.	1st Qu.	Med.	3rd Qu.	Max.	% F	%M
LARa	22	24.75	28	49.5	59	159	163	171.5	177	185	48	63.5	69.5	79.75	100	50	50
MotionSense	18	25	28	31.25	46	161	164.8	175.5	180	190	48	60	71	80.5	102	41.67	58.33
Mobiact	20	22.25	25	26	40	158	170	176	180	193	50	67	75.5	85	120	27.59	72.41
Sisfall	19	22.25	26.50	64	75	149	156	164	170	183	41.5	52.25	62	72	102	50	50
Sisfall Young	19	21	23	25	30	149	156.5	165	171	183	41.5	49.25	58.5	68.75	80.5	52.17	47.83

MoCap dataset consists of kinematic recordings of logistics activities. The combination of datasets for experimentation would bring forth the representation biases that may be present in the datasets, irrespective of their feature quantity.

All four datasets provide the age, gender, weight and height characteristics of the subjects. Table 2 presents the statistical analysis of the characteristics of the subjects. Furthermore, a sub-categorization of the Sisfall dataset, focusing on young subjects of the dataset, is presented. This subset is created due to incomplete data for older subjects within this dataset. Mobiact consists of the least variations in age and has more male subjects. The LARa, MotionSense, and Sisfall datasets are more varied in age distribution and nearly equal in gender distribution.

### 3.4 Experimental Design

The initial analysis showed a significant correlation between the height and weight characteristics to the gender of the individuals for all datasets. Table 3 shows the frequency of the different characteristic values, namely, height and weight, to gender. Given the small number of subjects across datasets, binary categorization based on the dataset median was considered to ensure sufficiently large training sets. Thus, height and weight are classified as Short/Tall and Light/Heavy. The table shows the division after combining all datasets, implying that including these characteristics in the selection of the subjects would essentially repeat the trend. As a result, we focus on the age and gender of the subjects to test the hypothesis. Age characteristics were divided into the levels ‘young’ and ‘old’, while binary categorization of gender (male and female) was followed as per the datasets. For the Sisfall dataset, we utilized the age divisions provided by the dataset creators. Thus, we have four combinations of

**Table 3.** Frequencies of gender and categorized weight and height for all datasets.

	Weight		Height	
	Light	Heavy	Short	Tall
Female	41	12	49	4
Male	27	56	18	65

**Table 4.** Description of the HM for the training set.

Group	Heterogeneity Measure
1	All subjects share the same characteristics
2	Subjects have two different characteristics (e.g. old women and young men are used in the training set)
3	Subjects have three different characteristic level combinations
4	All four different characteristics combination are included in the training set

the two characteristics, referred to as characteristic levels: young woman, old woman, young man, and old man. Following the evaluation strategy outlined in Section 3.2, the experiments encompass four different groups of heterogeneity, as depicted in Table 4.

The HM ‘2’ can be further divided into two subgroups depending on how the two types of characteristic levels differ: ‘2a’ refers to differences in one characteristic (e.g., young men and young women), and ‘2b’ refers to differences in both characteristics (e.g., young women and old men). Similarly, the HM ‘3’ consists of three different characteristic levels, for example, young man, old man, and young woman. Table 5 shows the frequency of the two characteristics under consideration in all data sets. The LARa dataset has an approximately similar number of subjects in the age and gender categories. However, clear differences in the number of subjects can be found in Mobiact. In accordance with the procedure described in Section 3.2, the number of subjects for the training of each dataset was determined. Specifically, four subjects were used to train the LARa, MotionSense, and Sisfall Young datasets. For the Mobiact dataset, the number of subjects was increased to 12, and for the Sisfall dataset, eight subjects were selected for training. The remaining subjects not included in the training sets were reserved for testing.

**Table 5.** Frequencies of gender and categorized age for all datasets.

	LARa		MotionSense		Mobiact		Sisfall		Sisfall Young	
	Young	Old	Young	Old	Young	Old	Young	Old	Young	Old
Female	5	3	7	3	7	9	12	7	6	6
Male	4	4	6	8	19	23	11	8	4	7

**Neural Networks and Training Procedure** This work uses three varied neural networks<sup>1</sup> for HAR; namely, two variations of time-series convolutional neural networks (CNN-IMU)- proposed by [14], a Long-Short Term Memory

<sup>1</sup> The code and parameters of the networks are available on [GitHub](#).

(LSTM) network and Transformer (Trans) proposed by [23]. The first variation of CNN-IMU has a block of four convolutional layers, two layers of multi-layer perceptron (MLP), and the softmax activation layer. Datasets with a few channels (less than 10) are trained on this network. The second CNN-IMU variant for high channel density has five blocks of four convolutional layers, followed by two layers of MLP. The LSTM network has four hidden layers of 256 dimensions, followed by two MLP layers and a softmax activation layer. Unlike classical classifiers, which require hand-crafted features, the deep learning architecture performs necessary feature extraction on the input data before the classification process during supervised learning. As a result, the method is robust against manual feature extraction biases.

The weights of the network are initialized using the orthogonal initialization method. The Cross-Entropy Loss function is utilized to calculate activity classification loss. The Root Mean Square Propagation (RMSProp) optimization is used with a momentum of 0.9 and weight decay of  $5 \times 10^{-4}$ . Gaussian noise with mean  $\mu = 0$  and standard deviation (SD)  $\sigma = 0.01$  is added to the sensor measurements to simulate sensor inaccuracies [14]. Dropout of probability  $p = 0.5$  was applied on the MLP, and early-stopping was implemented to avoid overfitting. The baseline architectures for each dataset were experimentally obtained post-hyperparameter search.

**Evaluation Metric** The accuracy (Acc) and weighted F1 score (wF1) were used to measure the activity metrics. wF1 was evaluated due to the unbalanced nature of the activity recordings in the datasets. Furthermore, recall and precision of the activity labels are evaluated.

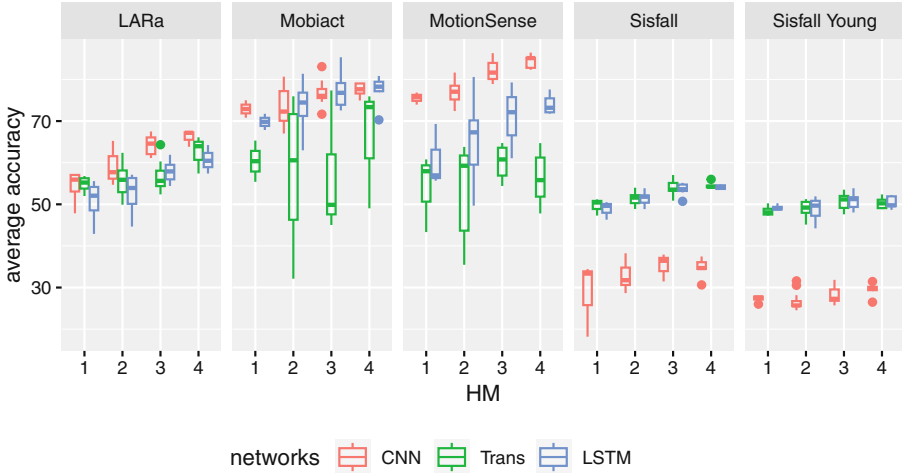
## 4 Experiments and Results

This section presents and analyses the results obtained from the experimental design discussed in Section 3.2. The first step in this direction is to achieve a baseline evaluation of the networks on the selected datasets to understand the networks' performance on a larger quantity of the same dataset, as shown in Table 6. CNNs and LSTMs perform well on the datasets. An exception is the case of Sisfall for CNN. In comparison, Transformer (Trans) perform poorly on all datasets except MotionSense. This can be associated with the training data quantity required for CNN-Transformers.

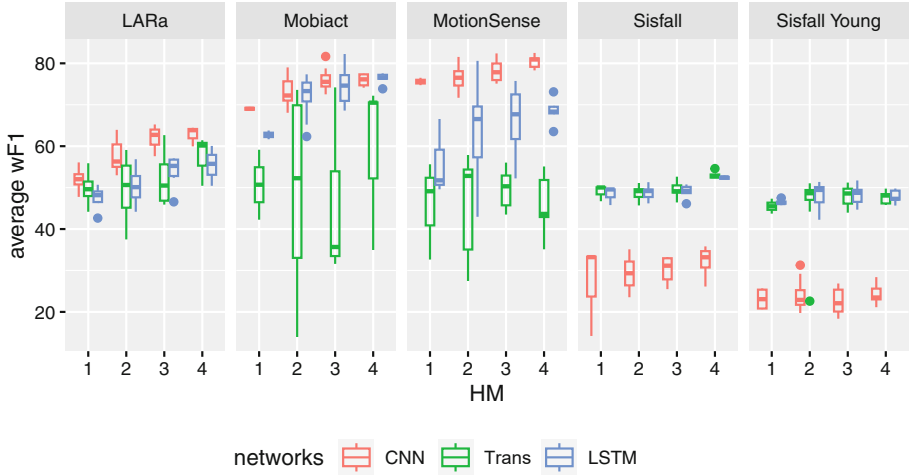
Next, the subjects of the training and validation sets are chosen based on the statistical hypothesis discussed in Section 3. The neural networks trained on the training sets created based on heterogeneity measures generally perform poorly, given the fewer data available in these sets. However, this work is focused on the comparative performance of the networks on the training sets as it is ensured that the sets have a similar quantity of data. Figure 2 and Figure 3 present the boxplots of accuracy and wF1-score for each HM group for all datasets. The networks are given a designated color and are followed for all plots in this work.

**Table 6.** Baseline implementation of neural networks on the selected datasets.

Dataset	Network	Batch Size	Epoch	Accuracy (%)	wF1 (%)
LARA	CNN-IMU	100	10	$88.0824 \pm 0.2149$	$87.5761 \pm 0.2497$
	LSTM	50	15	$82.5851 \pm 0.5524$	$81.6674 \pm 0.6279$
	Trans	100	30	$71.6814 \pm 12.9298$	$65.5721 \pm 20.0809$
Mobiact	CNN	50	30	$94.5179 \pm 0.1038$	$94.3257 \pm 0.1145$
	LSTM	50	15	$95.6331 \pm 0.1631$	$95.5188 \pm 0.1691$
	Trans	50	15	$71.4007 \pm 29.0897$	$65.7899 \pm 36.5785$
MotionSense	CNN	50	30	$95.9017 \pm 0.1381$	$95.8639 \pm 0.1278$
	LSTM	100	30	$96.0538 \pm 0.2132$	$96.0182 \pm 0.2079$
	Trans	100	15	$91.1124 \pm 0.6498$	$91.0656 \pm 0.6338$
Sisfall	CNN	50	50	$63.3665 \pm 0.8636$	$63.2645 \pm 0.8082$
	LSTM	50	50	$74.4645 \pm 0.3826$	$74.3861 \pm 0.3169$
	Trans	100	30	$71.2942 \pm 0.2988$	$70.8207 \pm 0.4622$

**Fig. 2.** Results on the average accuracy measured in percentage for all datasets on all HM groups.

The performance measures of all datasets present similar trends. In particular, the results show an increase in the average accuracy of the classification experiments, especially for datasets with large age differences (such as LARA and MotionSense). Interestingly, the increase is not significant when comparing the results of Sisfall, specifically Sisfall Young. However, Sisfall shows a large age variation within the dataset. A major point is that the variation in age is clustered rather than linear, the impact of which can be seen in the results of HM ‘3’ and ‘4’. A similar trend is visible with the wF1 values. The average wF1 of



**Fig. 3.** Results on the average wF1-Score measured in percentage for all datasets on all HM groups.

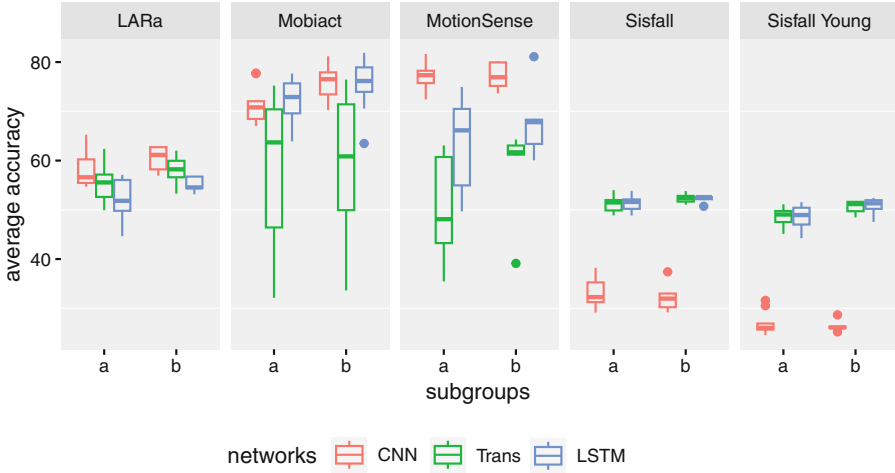
the classification experiments shows an increase in performance with an increase in heterogeneity measure in alignment with the hypothesis.

The influence of the different network architectures on the performance metrics shows clear differences between the datasets. Although the CNN architecture is competitive in datasets such as LARA, MotionSense and Mobiact, it performs comparatively worse in the two subsets of Sisfall. In particular, the Transformer network shows large variations in the values of the performance metrics in the Mobiact and MotionSense datasets, especially in the heterogeneity group ‘2’. However, for all networks and datasets, the standard deviation of accuracy and wF1 performances decreases with increased HM. This can be associated with the improved robustness of the network.

The performance difference in HM ‘2’ may be attributed to the different subgroups ‘2a’ and ‘2b’. As discussed previously, while HM ‘2a’ consists of variations of one characteristic of the training set subjects, HM ‘2b’ consists of variations of two characteristics. This means that the training data of HM ‘2b’ has more diversity in the scope of the subject’s physical characteristics than ‘2a’. In our case, at least two subjects in the training set are retained to have similar characteristics. Figure 4 presents the accuracy results of HM ‘2’ subgroups. Similar results for wF1 can be found in Figure S1 in the Supplement. Based on the trend seen in Figure 2 and Figure 3, an improvement in the performance measure would be expected with increasing heterogeneity in the training set. However, it is worth noting that no significant differences in performance can be observed, even if the average accuracy values are slightly higher for HM ‘2b’. A greater difference can be observed for MotionSense and Transformer in particular.

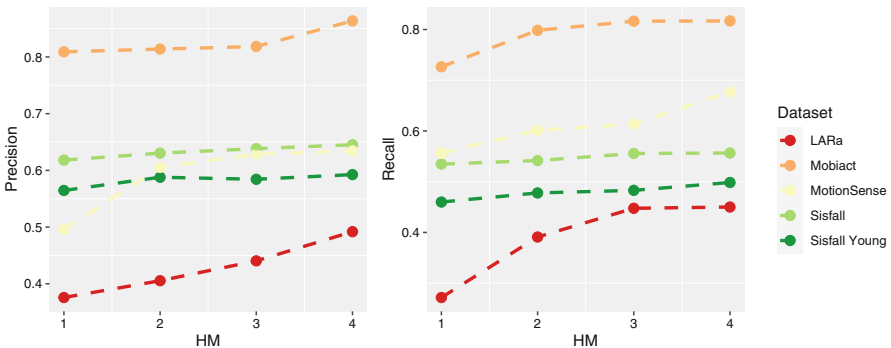
Turning to a more targeted analysis by examining the performance metrics for specific activity labels. Here, we focus on recall and precision to assess the





**Fig. 4.** Results on the average accuracy measured in percentage for all datasets on HM subgroups ‘2a’ and ‘2b’.

classifier’s ability to identify individual activities accurately. In this context, the focus is narrowed to the activity of ‘walking’ since it is consistently present across all datasets. Figure 5 shows the averaged precision (left) and recall (right) across all corresponding experiments of the HM groups. Figures S2 and S3 in the Supplement provide more detailed results. An increased precision and recall across higher HM groups is generally observed. However, recall is more receptive to an increase in characteristic levels. This trend aligns with our previous findings, suggesting that as heterogeneity in the training sample increases, the classifier’s ability to accurately identify the ‘walking’ activity improves on the unseen test data. Thus, this work with statistical evaluation proves that for the same amount of training data, having varied subject characteristics, here heterogeneity, helps enhance neural network performance on unseen test data.



**Fig. 5.** Mean precision and recall for ‘walking’ activity for all datasets.

## 5 Discussion

The experiments of this work aimed to answer the three research questions iterated in Section 1. Here, we answer the research questions based on the analysis of the experiments.

*RQ1: Do the physical characteristics of humans influence activity recognition performance?*

The experiments indicate that training data comprising diverse physical characteristics compared to a training set with homogeneous physical characteristics of the subjects improves accuracy on unseen testing data with subjects of varied physical attributes. Specifically, a systematic increase in the heterogeneity of training data while maintaining the quantity of training data of subjects improved classification accuracy. Thus proving the influence of human characteristics on the HAR classifier performance, answering *RQ1*.

*RQ2: What physical characteristics should be considered when selecting subjects to create a robust classifier?*

Noting that the majority of HAR datasets are limited in size and variability of subject characteristics, the datasets chosen as part of this work showed an inherent correlation between height and weight characteristics to gender. Within the datasets' and study's limitations, the experiments indicate that gender significantly influences the HAR models, followed by age, height and weight. Due to the unavailability of further physical characteristic information in HAR datasets and the correlation of height and weight to gender, the answer to *RQ2* is restricted mainly to age and gender characteristics. The study does not intend to discriminate based on these attributes but to identify representation biases. Thus, we recommend the creation of well-documented, large datasets with diverse subjects to further the research on physical characteristics that influence the robustness of HAR classifiers. For example, the influence of handedness and ethnicity.

Furthermore, we recommend that dataset creators ensure the presence of subjects with extreme characteristics in their dataset, along with a more significant number of subjects with diverse physical characteristics. However, as seen in Sisfall, a uniform selection of subjects from the range of a characteristic is ideal compared to clusters within the range of characteristics. To elaborate, it is essential to consider the variation of the characteristics within the subgroups, as was evident when comparing the age groups of the Sisfall dataset. This practice increases the classifier's robustness and contributes to its overall performance.

*RQ3: Can we develop a metric for representation bias in activity recognition classifiers?*

The answer to *RQ3*, on the development of a metric for representation bias for HAR classifiers, is that a metric is not ideal for the dataset curation process but rather an evaluation strategy focusing on a heterogeneity measure to curate training data for neural networks is ideal. The experiments clarified that HM directly impacts accuracy, wF1 and recall. Precision showed relatively less response to low variations in HM measures. However, more improvement in precision was found with maximum heterogeneity in the training data. As physical characteristic information is blended into the motion data recorded by sensors, a

significant limitation is the complexity of accurately measuring and quantifying representation bias and its interaction with other dataset biases. Thus, unless the intention is to classify the physical characteristics directly, an evaluation strategy is preferred than a metric.

## 6 Conclusions

This work aimed to evaluate representation biases in HAR systems by analyzing the impact of subjects' physical characteristics on classifier performance. Understanding these biases is crucial for developing more accurate, reliable and generalized HAR models and to guide the dataset creation process for novel HAR applications. To achieve this, we systematically curated training data for state-of-the-art HAR classifiers and evaluated their performance on four datasets with subjects of varying physical characteristics. Further, based on the experimental results, we answered the three main research questions the work focused on. The work established an influence of the subject characteristics on the performance of human activity recognition neural network models. Further, within the limitation of the subject characteristics made available in public HAR datasets, this study provided suggestions on the physical characteristics to focus on. Finally, the work provides a recommendation to HAR dataset creators on subject selection criteria for dataset creation based on the sequential science of experiments.

This work focused on binary classes within characteristics. In future work, evaluations on multiple sub-classes within each physical characteristic and the evaluation's impact must be performed to further generalize this contribution's conclusions. These may require extensive amounts of data to learn HAR through supervised learning. Thus, larger, well-documented datasets with variations in subjects' physical characteristics (multiple classes for each characteristic) are required to analyze these models. In addition, datasets consisting of detailed subject characteristics are desirable for identifying new dimensions of the dataset bias, such as the impact of handedness.

**Acknowledgment.** The work in this publication was supported by the German Federal Ministry of Education and Research (BMBF) in the context of the project "LAMARR Institute for Machine Learning and Artificial Intelligence" (LAMARR24B).

## References

1. Álvarez-Aparicio, C., Guerrero-Higueras, Á.M., González-Santamarta, M.Á., Campazas-Vega, A., Matellán, V., Fernández-Llamas, C.: Biometric recognition through gait analysis. *Sci. Rep.* **12**(1), 1–11 (2022)
2. Avsar, H., Altermann, E., Reining, C., Rueda, F.M., Fink, G.A., ten Hoppel, M.: Benchmarking Annotation Procedures for Multi-channel Time Series HAR Dataset. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (2021)

3. Balayn, A., Lofi, C., Houben, G.J.: Managing bias and unfairness in data for decision support: a survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. *VLDB J.* **30**(5), 739–768 (2021)
4. Bragança, H., Colonna, J.G., Oliveira, H.A., Souto, E.: How validation methodology influences human activity recognition mobile systems. *Sensors* **22**(6), 2360 (2022)
5. Chang, Y., Mathur, A., Isopoussu, A., Song, J., Kawsar, F.: A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **4**(1), 39:1–39:30 (Mar 2020)
6. Ferrari, A., Micucci, D., Mobilio, M., Napolitano, P.: Personalization in human activity recognition. arXiv preprint [arXiv:2009.00268](https://arxiv.org/abs/2009.00268) (2020)
7. van Giffen, B., Herhausen, D., Fahse, T.: Overcoming the pitfalls and perils of algorithms: A classification of machine learning biases and mitigation methods. *J. Bus. Res.* **144**, 93–106 (2022)
8. Hamidi, M., Osmani, A.: Human activity recognition: a dynamic inductive bias selection perspective. *Sensors* **21**(21), 7278 (2021)
9. Joshi, I., Grimmer, M., Rathgeb, C., Busch, C., Bremond, F., Dantcheva, A.: Synthetic data in human analysis: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024)
10. Karkkainen, K., Joo, J.: Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 1548–1558 (2021)
11. Khan, M.A.A.H., Roy, N., Misra, A.: Scaling Human Activity Recognition via Deep Learning-based Domain Adaptation. In: *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. pp. 1–9 (Mar 2018)
12. Lockhart, J.W., Weiss, G.M.: Limitations with activity recognition methodology & data sets. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. pp. 747–756. ACM, Seattle Washington (Sep 2014)
13. Malekzadeh, M., Clegg, R.G., Cavallaro, A., Haddadi, H.: Mobile sensor data anonymization. In: *Proceedings of the International Conference on Internet of Things Design and Implementation*. pp. 49–58. *IoTDI '19*, ACM, New York, NY, USA (2019)
14. Moya Rueda, F., Grzeszick, R., Fink, G.A., Feldhorst, S., Ten Hompel, M.: Convolutional neural networks for human activity recognition using body-worn sensors. In: *Informatics*. vol. 5, p. 26. Multidisciplinary Digital Publishing Institute (2018)
15. Nair, N.R., Moya Rueda, F., Reining, C., Fink, G.A.: Multi-channel time-series person and soft-biometric identification. In: *International Conference on Pattern Recognition*. pp. 256–272. Springer (2022)
16. Niemann, F., Reining, C., Moya Rueda, F., Nair, N.R., Steffens, J.A., Fink, G.A., ten Hompel, M.: LARa: Creating a Dataset for Human Activity Recognition in Logistics Using Semantic Attributes. *Sensors* (2020)
17. Nunes, A., Trappenberg, T., Alda, M.: The definition and measurement of heterogeneity. *Transl. Psychiatry* **10**(1), 299 (2020)
18. Reining, C., Nair, N.R., Niemann, F., Rueda, F.M., Fink, G.A.: A tutorial on dataset creation for sensor-based human activity recognition. In: *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. pp. 453–459. IEEE (2023)

19. Reining, C., Rueda, F.M., Niemann, F., Fink, G.A., ten Hompel, M.: Annotation Performance for multi-channel time series HAR Dataset in Logistics. In: 2020 IEEE PerCom Workshops (2020)
20. Riaz, Q., Vögele, A., Krüger, B., Weber, A.: One small step for a man: Estimation of gender, age and height from recordings of one step by a single inertial sensor. *Sensors* **15**(12), 31999–32019 (2015)
21. Selzler, R., Chan, A.D.C., Green, J.R.: Tsea: An open source python-based annotation tool for time series data. In: 2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA). pp. 1–6 (2021)
22. Shahbazi, N., Lin, Y., Asudeh, A., Jagadish, H.: Representation bias in data: a survey on identification and resolution techniques. *ACM Comput. Surv.* **55**(13s), 1–39 (2023)
23. Shavit, Y., Klein, I.: Boosting inertial-based human activity recognition with transformers. *IEEE Access* **9**, 53540–53547 (2021)
24. Sucerquia, A., López, J., Vargas-Bonilla, J.: SisFall: A Fall and Movement Dataset. *Sensors* **17**(12), 198 (Jan 2017)
25. Taha, K., Yoo, P.D., Al-Hammadi, Y., Muhaidat, S., Yeun, C.Y.: Learning a deep-feature clustering model for gait-based individual identification. *Computers & Security* **136**, 103559 (2024)
26. Vavoulas, G., Chatzaki, C., Malliotakis, T., Pedititis, M., Tsiknakis, M.: The mobiact dataset: Recognition of activities of daily living using smartphones. In: International conference on information and communication technologies for ageing well and e-health. vol. 2, pp. 143–151. SciTePress (2016)
27. Wang, A., Liu, A., Zhang, R., Kleiman, A., Kim, L., Zhao, D., Shirai, I., Narayanan, A., Russakovsky, O.: Revise: A tool for measuring and mitigating bias in visual datasets. *International Journal of Computer Vision* pp. 1–21 (2022)
28. Yang, Y., Gupta, A., Feng, J., Singhal, P., Yadav, V., Wu, Y., Natarajan, P., Hedau, V., Joo, J.: Enhancing fairness in face detection in computer vision systems by demographic bias mitigation. In: Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society. pp. 813–822 (2022)
29. Zhang, M., Li, H., Ge, T., Meng, Z., Gao, N., Zhang, Z.: Integrated sensing and computing for wearable human activity recognition with mems imu and ble network. *Measurement Science Review* **22**(4), 193–201 (2022)



# Secure Sleep Apnea Detection with FHE and Deep Learning on ECG Signals

Bharat Yalavarthi<sup>1</sup>(✉), Arjun Ramesh Kaushik<sup>1</sup>, Tilak Sharma<sup>1</sup>,  
Charanjit Jutla<sup>2</sup>, and Nalini Ratha<sup>1</sup>

<sup>1</sup> University at Buffalo, Buffalo, NY, USA  
{byalavar,kaushik3,tilaksha,nratha}@buffalo.edu

<sup>2</sup> IBM Research, Yorktown Heights, NY, USA  
csjutla@us.ibm.com

**Abstract.** Sleep apnea, a prevalent sleep disorder affecting individuals of all demographics, poses a threat of significant disruption to daily life. The analysis of Electrocardiogram (ECG) data facilitates the accurate diagnosis of sleep apnea. With the advent of machine learning and its accessibility through cloud services, doctors have been compelled to enhance their diagnostic capabilities by integrating deep learning into their analytical tools. However, challenges such as data privacy, security, and confidentiality regulations are hindering the adoption of deep learning in the healthcare domain. In this research, we address these challenges by proposing an end-to-end encrypted framework to analyze encrypted ECG signals and diagnose sleep apnea. Leveraging Fully Homomorphic Encryption (FHE) on deep learning models ensures privacy and security by design while enabling computations on encrypted data. To overcome the unique challenges posed by handling encrypted data in deep learning models, we introduce novel and efficient techniques for adapting several key components such as the convolutional layer, max pooling, ReLU activation, and fully connected layer to the FHE domain. Our approach includes adapting the convolutional layer in the spectral domain, implementing fully connected layers as generalized matrix multiplication, and employing approximation methods for ReLU activation and max pooling. The experimental results on real encrypted ECG data demonstrate the feasibility and efficacy of our proposed framework, achieving a remarkable accuracy of 99.56% in detecting sleep apnea. Our proposed encrypted network does not lose any predictive performance compared to its plaintext counterpart. This research underscores the potential of encrypted data processing in significantly enhancing the security and privacy of healthcare services, particularly in the domain of sleep apnea diagnosis.

**Keywords:** Convolutional Neural Networks · Fully Homomorphic Encryption · Homomorphic Fourier Transform · Sleep Apnea Detection

# 1 Introduction

Sleep apnea is a prevalent sleep disorder characterized by abnormal reductions or pauses in breathing during sleep, leading to inadequate oxygen supply to the patient [27]. The consequential impact on sleep quality can manifest in short-term issues such as low concentration, daytime sleepiness, and irritability, while long-term effects may include heart complications and diabetes <sup>1</sup>. Polysomnography (PSG) is the conventional diagnostic test for sleep apnea, yet its drawbacks, including time-consuming procedures and limited monitoring periods, necessitate the exploration of alternative methods [27]. Electrocardiogram (ECG) signals play a crucial role in data-driven diagnostic methods for a wide range of diseases. These signals provide detailed insights into the electrical activity of the heart, enabling the identification, monitoring and early detection of various conditions [8]. Moreover, ECG signals have been recognized as significant features in the detection of sleep apnea and also are cost-effective and convenient. By analyzing the variations and patterns in ECG data, advanced algorithms can detect anomalies indicative of sleep apnea with high accuracy, contributing to more effective and timely diagnosis and treatment of this prevalent disorder [24]. Several studies have demonstrated the effectiveness of ECG signals in automated sleep apnea detection [10], leveraging deep learning models for accurate, accessible, and continuous monitoring [17].

As cloud-based deep learning models gain popularity in medical diagnosis [16], the importance of ensuring data security and privacy becomes paramount [30]. Large-scale data breaches and identity theft underscore the challenges of constructing resilient and secure systems in the open environment of the Internet [12]. Given the sensitive nature of the medical diagnosis, strong data privacy regulations, and ECG signals containing personally identifiable information [14], end-to-end encryption is crucial when utilizing healthcare cloud services. Alternatives like confidential computing cannot ensure the same level of privacy as FHE systems as the data must be decrypted during the analysis phase, rendering it vulnerable. Additionally, it is susceptible to side-channel attacks, as demonstrated by various studies [26].

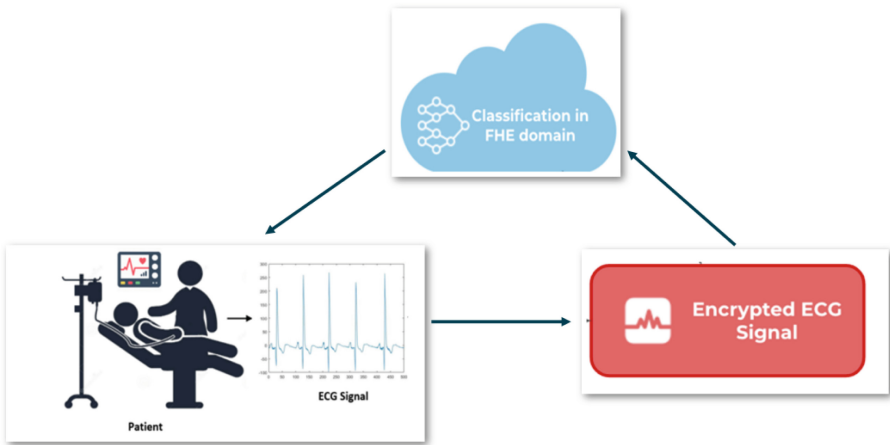
This paper proposes a convolutional neural network (CNN) classifier for sleep apnea detection using homomorphically encrypted ECG signals. The end-to-end secure framework ensures that the ECG signal remains encrypted throughout the process, preserving patient privacy. The large-scale availability of ECG data enabled CNN models to achieve near-perfect accuracy scores [17], making them competent for cloud-based disease diagnosis. The high-level framework of the proposed system is shown in Figure 1. The user collects the ECG signal, encrypts it using the private key, and sends it to the cloud service provider. After the inference is performed on the encrypted data, the diagnosis result (in encrypted form) is communicated to the user who only can decrypt it. We consider a comprehensive threat model for cloud diagnostic services, presuming a scenario where the cloud is compromised, and an attacker gains access to medical data,

---

<sup>1</sup> <https://houstonleepsolutions.com/what-is-sleep-apnea-and-do-i-have-it/>

including diagnostic results. Our FHE solution ensures that, even in the event of a cloud breach, the data remains secure as it is maintained in encrypted form, preventing unauthorized use by the attacker.

Furthermore, various data and computational adjustments are implemented for efficient inference of encrypted data. To address the high processing times associated with operations on encrypted data, particularly in convolutional layers, this research incorporates multi-threading techniques. By optimizing the distribution of filters across threads, we aim to achieve more efficient computations while minimizing unnecessary memory overhead. Our approach enhances the practicality of applying encrypted deep learning models in real-world scenarios. Displayed equations are centered and set on a separate line.



**Fig. 1.** Framework of the proposed privacy-preserving sleep apnea detector

## 2 FHE basics

A homomorphic encryption scheme is characterized as an encryption system in which a set of operations on plaintexts can be executed directly on the ciphertexts without the need for decryption. This capability is attained through addition and multiplication operations as these two operations collectively form a functionally complete set over finite rings [23]. Let  $pKey$  and  $sKey$  denote the public and secret keys, while  $Enc$  and  $Dec$  represent the encryption and decryption processes. Consider plaintext values  $pt1$  and  $pt2$ . Encrypting  $pt1$  and  $pt2$  using the public key  $pKey$  results in  $ct1 = Encrypt(pt1, pKey)$  and  $ct2 = Enc(pt2, pKey)$ , representing their encrypted forms. A cryptosystem is considered homomorphic concerning a chosen operator (eg: addition, multiplication), denoted as  $\circ$ , if there exists another operator  $\bullet$  such that  $pt1 \circ pt2 = Dec(ct1 \bullet ct2, sKey)$ .



It's crucial to emphasize the broad spectrum of homomorphic encryption, accommodating different types designed to meet diverse computational requirements. Partially Homomorphic Encryption (PHE) permits only addition or multiplication operations. Somewhat Homomorphic Encryption (SHE) enables restricted computation on ciphertexts. Leveled Homomorphic Encryption (LHE) facilitates computations on ciphertexts with limited depth, providing the option to increase depth through multiple encryption levels. Fully Homomorphic Encryption (FHE) allows computations on ciphertexts of any depth and complexity, making it the most flexible of the lot.

Fully Homomorphic encryption schemes like BGV and BFV, building upon the first-generation FHE systems, were aimed at enhancing computational efficiency through leveled structures. These systems introduce optimizations like re-linearization and modulus-switching. In 2017, a novel homomorphic encryption scheme emerged named CKKS. This scheme improves efficiency and expands applicability across various arithmetic applications. CKKS also enhanced the efficiency of BGV/BFV by enabling quicker numerical computation through approximation. [12].

Our work utilizes FHE based on the CKKS scheme to enable secure computation on encrypted ECG signal data. However, several trivial computational operators used in deep learning are yet to be implemented in the FHE framework without compromising security. **In this work, we develop FHE-compatible operators for ECG analysis using a fully learned deep learning network for inferencing.**

### 3 Related Work

In the realm of privacy-preserving disease detection and deep learning with FHE, prior research has made notable strides. [25] introduced a method for arrhythmia diagnosis, achieving 98% accuracy by leveraging Support Vector Machines (SVM) on encrypted ECG signals. [29] employed classical regression techniques to fit and perform inference on encrypted data for seizure detection and predicting predisposition to alcoholism using EEG signals. Additionally, [3] proposed a toolbox of statistical techniques for secure genome analysis using encrypted genetic data.

There have been alternate privacy-preserving techniques, with a significant focus on federated learning. [19] utilized federated learning for Alzheimer's disease detection, while [21] applied it to fMRI analysis. However, federated learning has inherent vulnerabilities, such as communication risks between nodes and the central agent, as well as the storage of data in plaintext, making it susceptible to potential breaches [19]. Additionally, [31] proposed a sleep apnea monitoring mechanism employing fog computing to enhance security but several studies showed its vulnerability to potential man-in-the-middle attacks.

In the context of adapting Convolutional Neural Networks (CNNs) to FHE, various studies have been conducted. [15] introduced a 2D CNN in FHE for inference on MNIST and Melanoma datasets using spatial convolution. [1] explored

accelerating CNN inference in FHE using GPUs on MNIST and CIFAR-10 datasets. Notably, [22], [20], and [2] suggested performing convolution in the frequency domain to reduce the number of homomorphic operations. However, these methods exhibit limitations such as the absence of strided convolution, incomplete adaptation of ReLU and max pooling layers to FHE, or the need for intermediate re-encryption or interactions with the client.

In contrast, our work addresses these gaps by incorporating strided convolution in the frequency domain, providing accurate adaptations for max pooling and ReLU to FHE, and eliminating the necessity for intermediate interactions with the client, bootstrapping, or re-encryption. These advancements distinguish our methodology from existing approaches, enhancing the efficiency and security of deep learning in the FHE domain.

## 4 Proposed Approach

The network architecture used for sleep apnea detection is shown in Figure 2. The key modules that are developed in FHE are (i) Convolution Layer; (ii) ReLU; (iii) Max pooling layer; and (iv) Fully connected layer.

ECG data from the University College Dublin Sleep Apnea Database was used in this work [11]. This dataset comprises complete overnight simultaneous three-channel Holter ECG recordings, featuring adult subjects exhibiting indications of sleep-disordered breathing. Each second within this recording period was labeled as either apneic or non-apneic by experts, thereby providing granular and comprehensive data for analysis comprising 8,05,926 training samples. The network is initially trained using plaintext training data from this dataset and the trained weights are used for inference on the encrypted test data.

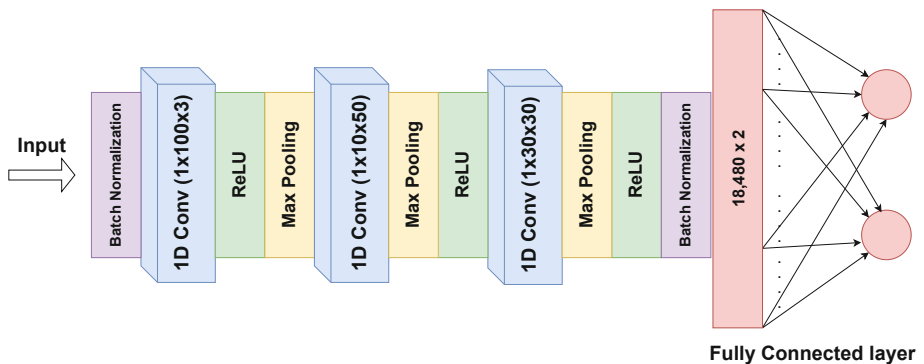
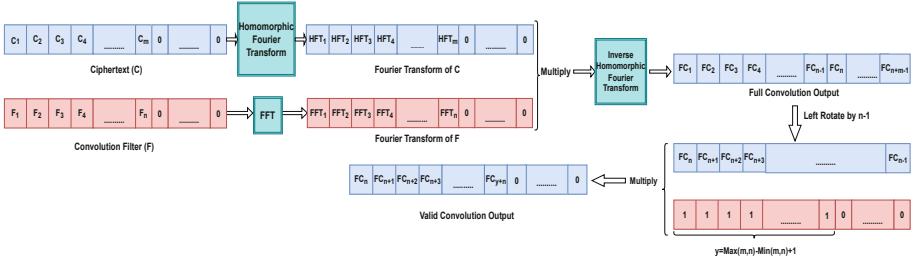


Fig. 2. CNN architecture used for secure sleep apnea detection.

We utilize the HEAAN library[6], which utilizes the CKKS scheme for FHE operations.



**Fig. 3.** 1D Convolution in FHE Domain. Input ciphertext and weights are multiplied in the frequency domain to obtain full convolution. Valid convolution output is obtained by rotating the full convolution by n-1 and extracting the valid convolution.

### 4.1 Adaptations for FHE

Computation within the FHE domain imposes various substantial constraints, including the absence of individual element access in encrypted arrays, restricted computation depth, high time complexity, and a lack of native support for fundamental operators such as a comparator. In this section, we discuss the adaptations made to the data and the training process of the CNN to ensure compatibility with FHE.

Each Electrocardiogram (ECG) signal in the dataset comprises 1408 samples. Given that the HEAAN library supports the encryption of data with sizes that are powers of 2, we pad each input signal with zeros to extend it to a length of 2048. This extended, padded input signal is then encrypted into a single ciphertext. Consolidating the entire input signal into a single ciphertext is crucial for the efficiency of arithmetic operations on ciphertexts, leveraging Single Instruction, Multiple Data (SIMD) operations supported by HEAAN. To facilitate efficient arithmetic operations between ciphertext inputs and plaintext weights/filters, we also pad the latter with zeros, extending them to a length of 2048. It’s noteworthy that increasing the input size from 1408 to 2048 doesn’t introduce noticeable computational overhead due to the SIMD nature of operations in HEAAN. As we need to use approximate versions of ReLU and max pooling in FHE, we employ these approximations during the training of the plaintext model. This enables the model to adjust to these approximations during inference in the FHE domain, thereby not affecting the predictive performance.

### 4.2 Convolutional Layer

For computational efficiency, we realize convolution by Hadamard product of signal and filter in the frequency domain based on equations 1 and 2.

$$y[n] = \mathcal{F}^{-1} \{X(k) \cdot W(k)\} \tag{1}$$

where  $\mathcal{F}^{-1}$  is the inverse Fourier transform,  $y[n]$  is the convolution output at index n,  $X(k)$ ,  $W(k)$  is the Discrete Fourier transform (DFT) of the signal, and filter respectively at index  $k$ , Discrete Fourier transform for input  $x$  is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \frac{2\pi}{N} nk} \quad (2)$$

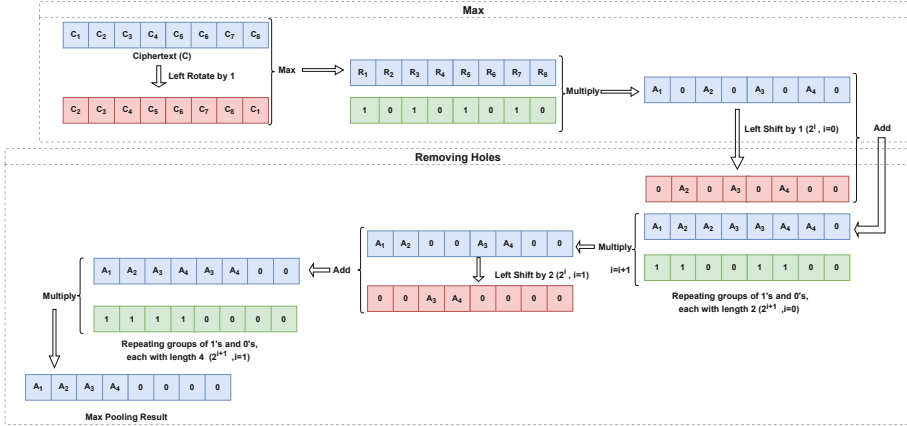
where  $X[k]$  is the DFT coefficient at frequency bin  $k$ ,  $x[n]$  is the input signal value at time index  $n$ , and  $N$  is the Length of the input signal.

Computing the DFT of encrypted data using standard plaintext methods is very time inefficient and consumes large multiplicative depth which is bad for FHE mapping. Homomorphic Fourier transform is used for computing the DFT of the encrypted data. Homomorphic Fourier transform as described in [13] uses Cooley-Tukey matrix factorization to construct an efficient algorithm for computing 1-D DFT of encrypted data. We observed that this algorithm is faster by around 165 times than regular matrix multiplication for computing 1-D DFT for an input size of 2048 as shown in Table 1.

**Table 1.** Time taken (in s) to calculate DFT in FHE

Input Size	Multiplication	Homomorphic Fourier transform
128	29.97	2.48
256	110.36	2.235
512	213.53	4.48
1024	435.60	5.1
2048	855.92	5.2
4096	1795.5	6.82

Standard Fast Fourier transform is used to transform the plaintext filter into the frequency domain. The result of the Hadamard product between the input and filter in the frequency domain followed by inverse DFT gives the full convolution output [18]. To get the valid convolution output, we rotate the resultant ciphertext by  $n-1$ , where  $n$  is the size of the filter. Subsequently, we perform a multiplication with an array that consists of alternating groups of 1's and 0's, as illustrated in Figure 3. Since our network has convolution layers with strided ( $>1$ ) convolution we devised a generic method to obtain an arbitrary strided convolution from frequency domain convolution output. As this output is a convolution with stride one, it is necessary to eliminate recurring patches of values. These patches have lengths corresponding to the stride (1, 2, 3, etc., for strides 2, 3, 4, etc.). The process involves multiplying the ciphertext by a plaintext vector with a specific pattern of 1s and 0s based on the patch size. Subsequently, the "remove holes" function, discussed in the following section, is invoked to complete this operation.



**Fig. 4.** Max Pooling illustration for Ciphertext of size 8. Max: Approximate max is applied to input and its left rotated variant and the result is multiplied by a plaintext array of alternating 1s and 0s to replicate the stride of two. Remove Holes: The result of the max stage contains alternating zeros which are all grouped and moved completely to the right.

### 4.3 Max Pooling

The pooling layer within our network employs max pooling with a kernel size of two and a stride of two. Due to the lack of support for the comparison operation in FHE schemes, we utilize an approximate max operation proposed by [5]. The formula for the approximate max value is given by:

$$\text{Max}(a, b; d) = \frac{(a + b)}{2} + \frac{\text{Sqrt}((a - b)^2; d)}{2}$$

Here,  $d$  represents the number of iterations used for computing the approximate square root, as also suggested in [5]. In our case, the two inputs  $a$  and  $b$  correspond to the input and a copy of the input left-rotated by 1. The result from the max function introduces alternating zeros or "holes," which need to be removed to get the pooling output. This poses a challenge due to the lack of access to individual elements.

To address this, we have devised a generic iterative process for hole removal in ciphertexts. This process involves left-shifting the ciphertext by  $2^i$ , adding it to the original, and then multiplying the result by an array containing repeating groups of 1's and 0's, each with a length of  $2^{i+1}$ . Here,  $i$  ranges from 0 to  $\log_2(N) - 2$  (where  $N$  is the length of the ciphertext). The entire process of max pooling is visually represented in Figure 4.

### 4.4 ReLU

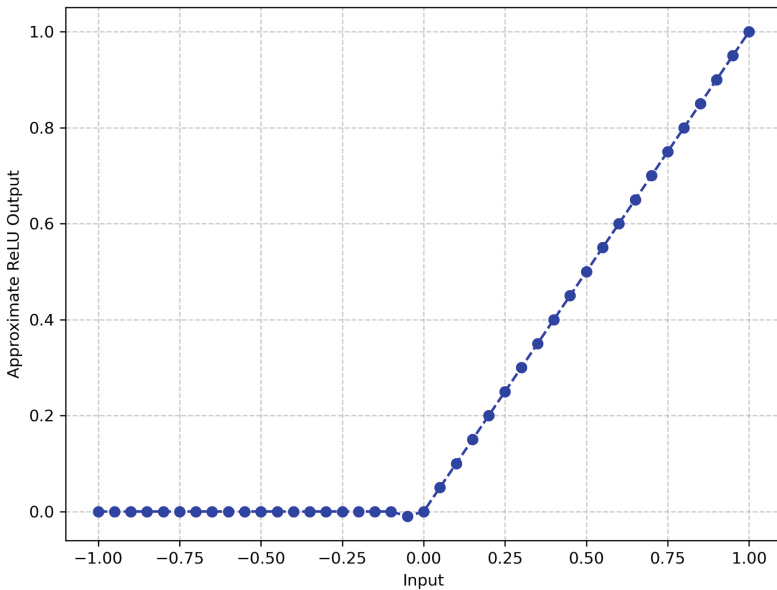
Homomorphic encryption schemes, including the HEAAN library, lack direct support for the comparison function. To address this, polynomial approximate

comparison is employed, as described in [5]. An asymptotically optimal comparison method named CompG, proposed by [7], approximates a sign function using composite polynomial approximation. CompG operates effectively within the input range of -1 to 1.

To ensure that the input to the Rectified Linear Unit (ReLU) falls within this specified range, output values from convolutional layers are normalized using a scaling factor before applying ReLU. The scaling factor is determined by the formula:

$$\text{Scaling Factor} = \max(|\text{maxValue}|, |\text{minValue}|)$$

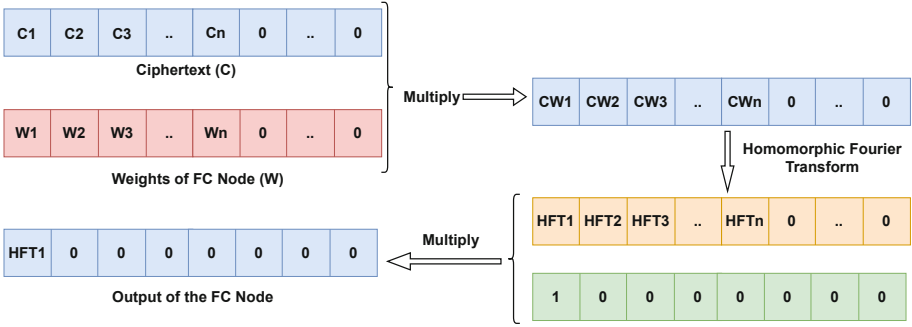
Here,  $|\text{maxValue}|$  and  $|\text{minValue}|$  represent the absolute maximum and minimum input values, respectively, observed for the corresponding ReLU block in the trained model on the training data. After applying ReLU, the reciprocal of the scaling factor is used to restore the original values of the positive inputs. For the ReLU implementation, the composite approximation technique is employed to compare the input value  $a$  against 0. This comparison function yields a result of 1 if  $a$  is greater than 0, 0 if  $a$  is less than 0, and 0.5 if  $a$  is equal to 0. The ReLU result is obtained by multiplying the output of the comparison function by the input value whose results are depicted in Figure 5.



**Fig. 5.** FHE ReLU results obtained using the approximate polynomial comparator

#### 4.5 Fully Connected Layer

In this layer, an array of ciphertexts serves as input. For each output node, element-wise multiplication is performed between each ciphertext in the array

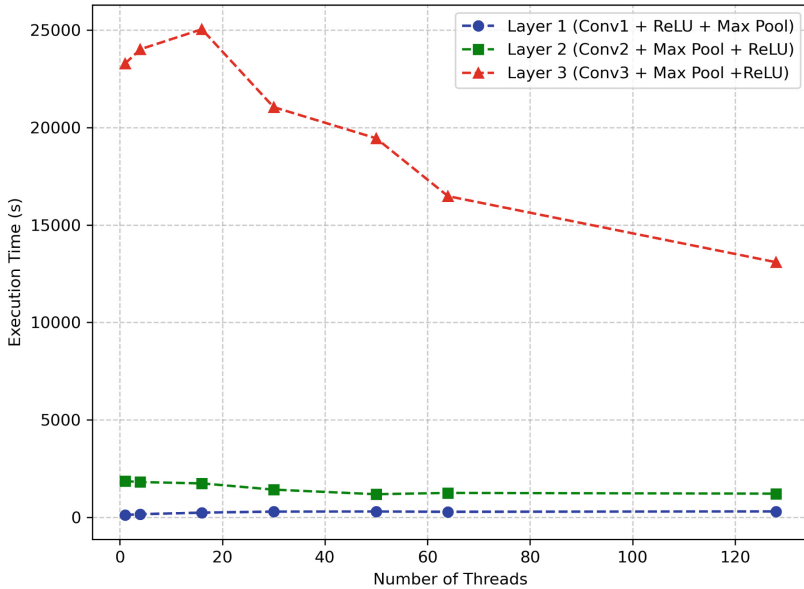


**Fig. 6.** FC Layer output computation illustration for single ciphertext input. Element-wise multiplication between ciphertext and weights is followed by calculating DFT. The final output for the FC node or weighted sum is the DC component of the DFT

and the corresponding node weights and the results are summed up. The outcome is a ciphertext whose elements need to be summed to obtain the final output of the node. To achieve the sum of these elements, a common approach involves left-shifting the ciphertext  $N-1$  times and iteratively adding it to the original ciphertext after each shift. However, this requires  $N-1$  rotations of the ciphertext, which can be computationally expensive. Instead, we employ a more efficient technique wherein we utilize the Discrete Fourier Transform (DFT) of the ciphertext to obtain the sum. The first element of the DFT represents the sum of all elements in the signal. The mapping of the fully connected layer to the FHE domain is depicted in Figure 6. Given that the problem at hand is binary classification, the fully connected layer outputs only two nodes. To determine the classification output of the network, the CompG comparator function is used to identify the higher activation value among these two nodes. The comparator function returns a ciphertext that can be used to find the predicted class label of the network.

### 4.6 Parallelization

To parallelize the convolutional layers, we employ the NTL multi-threading, which automatically manages thread creation and assignment in a manner that optimizes efficiency. However, not all layers are equally amenable to parallelization. Given the limited number of filters in the first layer (only three), we refrain from parallelization supported by experiments showing that using a single thread has the lowest latency. For the second layer, we find that using the threads equal to the number of filters in it is optimal while for the third layer using all the available 128 threads was found to be optimal as shown in Figure 7.



**Fig. 7.** Processing time taken by each convolutional layer for different numbers of threads

## 5 Results and Discussion

### 5.1 Performance evaluation

We randomly selected a few samples from the testing set and performed inference on the proposed encrypted network to evaluate its performance. Table 2 shows the classification performance of the encrypted model and Table 3 shows the layer wise error in the FHE domain. **The comparison between the encrypted network final classification output and the plaintext counterpart revealed no errors while providing 128 bit security, highlighting the efficacy of our adaptation to the FHE domain.**

### 5.2 Complexity Analysis

Table 4 shows the complexity analysis of our deep learning operators in the FHE domain. When separated by operations, fully connected layer was most expensive

**Table 2.** Predictive performance of the proposed model in FHE.

Performance Metric	Value
Accuracy	99.50%
Sensitivity	97.10%
Specificity	99.10%



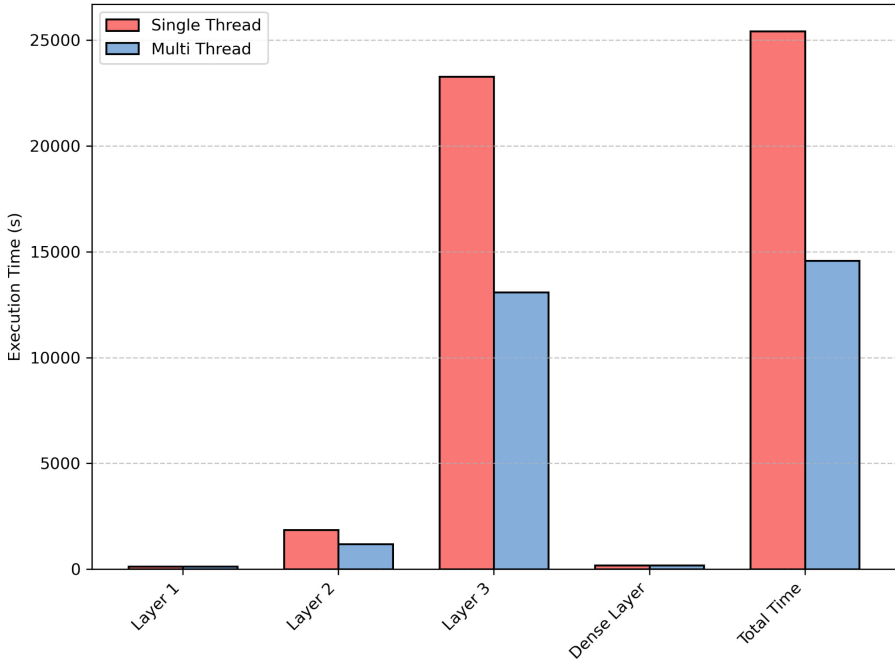
**Table 3.** Mean Average Error (MAE) induced due to FHE Computations

Layer	MAE
Layer 1 (Batch Norm + Conv1 + ReLU + Max Pooling)	0.010
Layer 2 (Conv2 + Max Pooling + ReLU)	0.102
Layer 3 (Conv3 + Max Pooling + ReLU+ Batch Norm)	0.011
Fully Connected Layer + Argmax ( <b>Final Output</b> )	<b>0</b>

followed by max pooling owing to its max operation. Although convolution was taking relatively lesser time, the number of convolution operations present in network is significantly higher than any other operations. Figure 8 shows the average time taken to process the data through each layer. A large chunk of the total time taken for inference is consumed by the third convolution layer consisting of 1500 convolutions. Our experimental results indicate a 36% drop in processing times for the second convolutional layer and a 44% drop for the third layer following parallelization. The layer’s processing times exhibit a less pronounced improvement than expected, indicating that additional factors like available memory may be affecting its performance. Using parallelization we were able to reduce the total inference time from 25430.2 s to 14571.25 s, attaining a speedup of 42.70%.

**Table 4.** Complexity Analysis (PT: Plaintext, CT: Ciphertext, HOP: Homomorphic Operations)

Operation	Conv	Max Pool	ReLU	Fully Connected Layer
#Additions	8	43	15	400
#PT-CT Muls	46	36	11	46
#CT-CT Muls	0	31	1	0
#CT Rotations	27	1	0	26
#HOPs	81	111	27	472
Latency (s)	20.70	25.14	17.14	517.49



**Fig. 8.** Processing time taken by each layer during inference in FHE domain

### 5.3 Advantages of FHE over other Privacy Preserving Methods

In our proposed approach we set the FHE parameters in HEAAN to obtain 128-bit security thereby providing cryptographic privacy guarantees throughout the detection process. It is well-established that other widely used privacy-preserving methods, such as federated learning or differential privacy, cannot guarantee such a high level of security [28]. Unlike other encryption methods like RSA, AES, that require data to be decrypted for processing, FHE ensures that data remains encrypted at all times making it end-to-end secure (Table 5). This significantly reduces the risk of data breaches and unauthorized access, helping to meet stringent regulatory requirements for data protection in healthcare, such as HIPAA in the United States. Privacy obtained through differential privacy involves compromising model performance for security, whereas FHE was able to maintain model performance while still providing a higher level of security [9] [4].

Moreover, the privacy use case we address in this work is a cloud-based disease detector. To provide maximum privacy, it is essential that patient data remains secure during transmission to the cloud, model inference, and transmission of the diagnosis results back to the patient. FHE is the only privacy-preserving technique that can provide end-to-end cryptographic privacy guarantees and security. Federated learning cannot protect against cloud breaches or ensure privacy from the cloud service provider. Differential privacy techniques cannot ensure

**Table 5.** Privacy and Security comparison of FHE and other methods

Phase	FHE	FL/DP/Encryption/etc..
Input Transmission to Cloud	Encrypted and Secure	Encrypted and Secure
Processing/Inference	Encrypted and Secure	Unencrypted and Unsecure
Diagnosis Result back to User	Encrypted and Secure	Encrypted and Secure

privacy during input/output transmission or processing and cannot guarantee privacy from the cloud service provider [9]. Even when these techniques are used in combination with regular encryption for data transmission, data still needs to be decrypted during processing leaving it vulnerable to attack as detailed in Table 5.

## 6 Conclusion and Future Work

In this paper, we propose the first end-to-end encrypted sleep apnea detector using deep neural networks. By employing FHE for encryption, we achieve 128-bit security for the entire pipeline of cloud-based medical diagnosis, including during inference. The proposed encrypted model detects sleep apnea with an accuracy of 99.50%, a specificity of 99.56%, and a sensitivity of 97.10%. We successfully adapted convolutional, fully connected, max pooling, and ReLU blocks of the CNN to the FHE domain. Specifically, we utilized the homomorphic Fourier transform to perform convolutions, employed approximate methods for executing ReLU and max pooling operations, and developed a novel technique to efficiently implement fully connected layers in the FHE domain.

For inference, we demonstrate that the encrypted model does not suffer any predictive performance loss compared to the plaintext version, thereby illustrating the feasibility of FHE-based systems in cloud-based medical diagnosis. Although our approach provides strong security guarantees and does not trade off performance for security, the drawback lies in the inference time, which we partially addressed through parallelization. Future directions for our work include developing more efficient parallelization strategies, and ciphertext packing schemes to further reduce the inference time.

## References

1. Badawi, A.: Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. *IEEE Trans. Emerg. Top. Comput.* **9**(3), 1330–1343 (2020)
2. Bian, S., Wang, T., Hiromoto, M., Shi, Y., Sato, T.: Ense: Efficient secure inference via frequency-domain homomorphic convolution for privacy-preserving visual recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9403–9412 (2020)

3. Blatt, M., Gusev, A., Polyakov, Y., Goldwasser, S.: Secure large-scale genome-wide association studies using homomorphic encryption. *Cryptology ePrint Archive*, Paper 2020/563 (2020). <https://doi.org/10.1073/pnas.1918257117>, <https://eprint.iacr.org/2020/563>, <https://eprint.iacr.org/2020/563>
4. Chamikara, M.A.P., Bertók, P., Khalil, I., Liu, D., Camtepe, S.: Privacy preserving face recognition utilizing differential privacy. *CoRR abs/2005.10486* (2020), <https://arxiv.org/abs/2005.10486>
5. Cheon, J., Kim, D., Kim, D., Lee, H.h., Lee, K.: Numerical Method for Comparison on Homomorphically Encrypted Numbers, pp. 415–445 (11 2019). [https://doi.org/10.1007/978-3-030-34621-8\\_15](https://doi.org/10.1007/978-3-030-34621-8_15)
6. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) *Advances in Cryptology - ASIACRYPT 2017*, pp. 409–437. Springer International Publishing, Cham (2017)
7. Cheon, J.H., Kim, D., Kim, D.: Efficient homomorphic comparison methods with optimal complexity. *Cryptology ePrint Archive*, Paper 2019/1234 (2019), <https://eprint.iacr.org/2019/1234>, <https://eprint.iacr.org/2019/1234>
8. De Marco, F., Ferrucci, F., Risi, M., Tortora, G.: Classification of qrs complexes to detect premature ventricular contraction using machine learning techniques. *PLOS ONE* **17**(8), 1–19 (08 2022). <https://doi.org/10.1371/journal.pone.0268555>, <https://doi.org/10.1371/journal.pone.0268555>
9. El Ouadrhiri, A., Abdelhadi, A.: Differential privacy for deep and federated learning: A survey. *IEEE access* **10**, 22359–22380 (2022)
10. Faust, O., Acharya, U.R., Ng, E., Fujita, H.: A review of ecg-based diagnosis support systems for obstructive sleep apnea. *Journal of Mechanics in Medicine and Biology* **16**, 1640004 (25 pages) (02 2016). <https://doi.org/10.1142/S0219519416400042>
11. Goldberger, A.L.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **101**(23), e215–e220 (June 13)
12. Gorantala, S., Springer, R., Gipson, B.: Unlocking the potential of fully homomorphic encryption. *Commun. ACM* **66**(5), 72–81 (apr 2023). <https://doi.org/10.1145/3572832>, <https://doi.org/10.1145/3572832>
13. Han, K., Hhan, M., Cheon, J.H.: Improved homomorphic discrete fourier transforms and the bootstrapping. *IEEE Access* **7**, 57361–57370 (2019). <https://doi.org/10.1109/ACCESS.2019.2913850>
14. Ingale, M., Cordeiro, R., Thentu, S., Park, Y., Karimian, N.: Ecg biometric authentication: A comparative analysis. *IEEE Access* **8**, 117853–117866 (2020). <https://doi.org/10.1109/ACCESS.2020.3004464>
15. Jain, N., Nandakumar, K., Ratha, N.K., Pankanti, S., Kumar, U.: Efficient CNN building blocks for encrypted data. *CoRR abs/2102.00319* (2021), <https://arxiv.org/abs/2102.00319>
16. Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., Wang, Y.: Artificial intelligence in healthcare: past, present and future **2**(4), 230–243 (2017). <https://doi.org/10.1136/svn-2017-000101>
17. John, A., Cardiff, B., John, D.: A 1d-cnn based deep learning technique for sleep apnea detection in iot sensors. *CoRR abs/2105.00528* (2021), <https://arxiv.org/abs/2105.00528>
18. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1d convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **151**, 107398 (2021). <https://doi.org/10.1016/j.ymssp.2020.107398>, <https://www.sciencedirect.com/science/article/pii/S0888327020307846>

19. Li, J., Meng, Y., Ma, L., Du, S., Zhu, H., Pei, Q., Shen, X.: A federated learning based privacy-preserving smart healthcare system. *IEEE Trans. Industr. Inf.* **18**(3), 2021–2031 (2022). <https://doi.org/10.1109/TII.2021.3098010>
20. Li, S., Xue, K., Zhu, B., Ding, C., Gao, X., Wei, D., Wan, T.: Falcon: A fourier transform based approach for fast and secure convolutional neural network predictions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8705–8714 (2020)
21. Li, X., Gu, Y., Dvornek, N., Staib, L.H., Ventola, P., Duncan, J.S.: Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results. *Med. Image Anal.* **65**, 101765 (2020). <https://doi.org/10.1016/j.media.2020.101765>, <https://www.sciencedirect.com/science/article/pii/S1361841520301298>
22. Lou, Q., Lu, W.j., Hong, C., Jiang, L.: Falcon: Fast spectral inference on encrypted data. *Advances in Neural Information Processing Systems* **33**, 2364–2374 (2020)
23. Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F.H., Aaraj, N.: Survey on fully homomorphic encryption, theory, and applications. *Proc. IEEE* **110**(10), 1572–1609 (2022)
24. McCausland, C., Biglarbeigi, P., Bond, R., Yadollahikhales, G., Finlay, D.: Time-frequency ridge analysis of sleep stage transitions. In: *2022 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. pp. 1–5 (2022). <https://doi.org/10.1109/SPMB55497.2022.10014897>
25. Miao, G., Ding, A.A., Wu, S.S.: Real-time disease prediction with local differential privacy in internet of medical things. *CoRR abs/2202.03652* (2022), <https://arxiv.org/abs/2202.03652>
26. Mulligan, D.P., Petri, G., Spinale, N., Stockwell, G., Vincent, H.J.M.: Confidential computing—a brave new world. In: *2021 International Symposium on Secure and Private Execution Environment Design (SEED)*. pp. 132–138 (2021). <https://doi.org/10.1109/SEED51797.2021.00025>
27. Peppard, P., Young, T., Barnet, J., Palta, M., Hagen, E., Hla, K.: Increased prevalence of sleep-disordered breathing in adults. *American journal of epidemiology* **177** (04 2013). <https://doi.org/10.1093/aje/kws342>
28. Podschwadt, R., Takabi, D., Hu, P., Rafiei, M.H., Cai, Z.: A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption. *IEEE Access* **10**, 117477–117500 (2022)
29. Popescu, A., Branea, I., Nita, C., Vizitu, A., Robert, D., Suciuc, C., Itu, L.: Privacy preserving classification of eeg data using machine learning and homomorphic encryption. *Applied Sciences* **11**, 7360 (08 2021). <https://doi.org/10.3390/app11167360>
30. Price, W., Cohen, I.: Privacy in the age of medical big data. *Nature Medicine* **25** (01 2019). <https://doi.org/10.1038/s41591-018-0272-7>
31. Ravikumar, G., Venkatachalam, K., AlZain, M.A., Masud, M., Abouhawwash, M.: Neural cryptography with fog computing network for health monitoring using iomt. *Comput. Syst. Sci. Eng.* **44**(1), 945–959 (2023)



# Efficient Convolution Operator in FHE Using Summed Area Table

Bharat Yalavarthi<sup>1</sup>(✉), Charanjit Jutla<sup>2</sup>, and Nalini Ratha<sup>1</sup>

<sup>1</sup> University at Buffalo, Buffalo, NY, USA  
{byalavar, nratha}@buffalo.edu

<sup>2</sup> IBM Research, Yorktown Heights, NY, USA  
csjutla@us.ibm.com

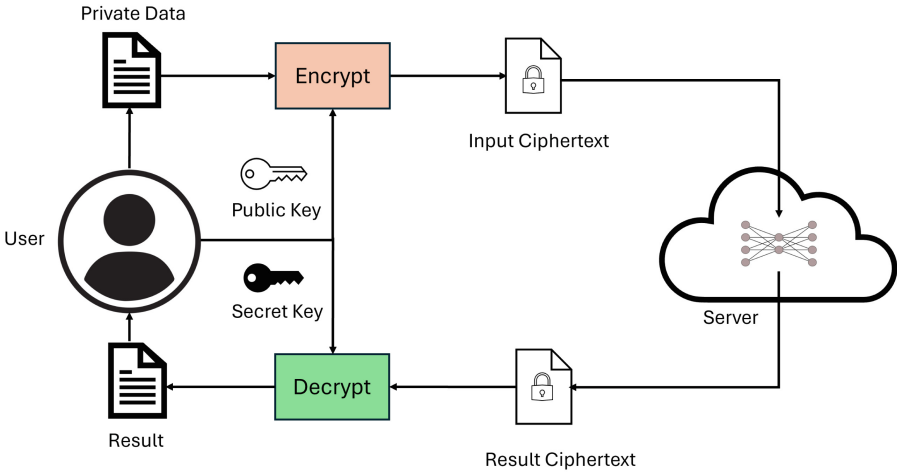
**Abstract.** To enhance privacy in Convolutional Neural Network (CNN) based inference methods, fully homomorphic encryption (FHE) is a golden tool. However, high latency and limited multiplicative depth are major problems in building CNNs for FHE. Convolution operations dominate the inference time of CNNs in FHE schemes due to the large number of costly multiplications and accumulation operations required. All the prior works have performed convolution in either the spatial or frequency domain. Alternatively, in this paper, we propose to use a summed area table (SAT) along with kernels approximated with box filters for the computation of convolution in 1D, 2D, and 3D space. The usage of box filters allows us to reduce the number of costly multiplications required to compute convolution. We show that the proposed method computes convolution output with lower latency than the standard spatial convolution method and can be applied with arbitrary kernels. We also show that the speed-up provided by our approach increases with the size of the image or kernel. Through the usage of SATs and box filters, we reduce the number of expensive multiplication operations required in convolution by 20%-52% and latency by 15%-89%.

**Keywords:** Convolution · Fully Homomorphic Encryption · Summed Area Tables · CNN

## 1 Introduction

The Fully Homomorphic Encryption (FHE) scheme provides a path for end-to-end secure and private inference of deep learning models. FHE finds its need in several applications where clients send sensitive information to the server for analysis by machine learning models as it enables computations on encrypted data [9]. Figure 1 shows the application of FHE in preserving the privacy of client data in cloud inferences. Several inherent limitations of FHE including restricted arithmetic operation support (only Addition and Multiplication), limited multiplicative depth, and high latency limit its practicality for real-world deployment. Convolutional Neural Networks (CNNs) have become the standard

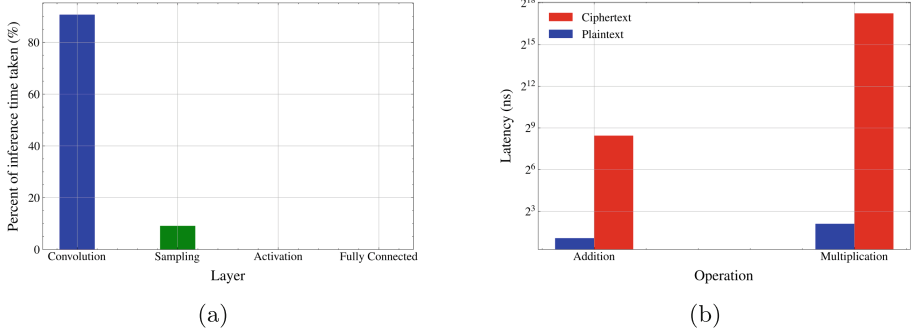
architecture used for solving many problems in computer vision [2], making it crucial to secure the data passed through CNNs for inference. In this work, we address the problem of latency of CNN inference in the FHE domain. [7] showed that convolution layers take around 90% of the inference time in CNNs (Figure 2a). In FHE there are a limited number of multiplications we can perform before the accumulating noise makes the ciphertext unrecoverable and a costly operation called bootstrapping is required to recover it partially. Moreover, multiplication in FHE is slower than in unencrypted domain by a factor of 10,000 [12] and addition is around 500 times faster than multiplication in FHE (Figure 2b). These factors strongly motivate us to reduce the number of multiplications in convolution to decrease the latency of CNNs in FHE. We propose a method using Summed Area Tables, and box filters to reduce the multiplications in convolution operation by replacing them with additions and thereby reducing the latency.



**Fig. 1.** Overview of using FHE to secure the client data when using cloud machine learning services.

Standard convolution is performed in either spatial or frequency domain. Spatial convolution involves applying a kernel/filter (used interchangeably) to an input signal or image by sliding it across the entire input and computing the weighted sum of the filter coefficients and the corresponding input values at each location (Fig. 3). In Frequency domain convolution the input and the filter are first converted into frequency domain using Fourier transform, the corresponding frequency representations are multiplied and are converted back to spatial domain using inverse Fourier transform.

A homomorphic encryption is defined as an encryption system where a set of operations on plaintexts can be performed directly on the corresponding ciphertexts. Let  $pKey$  and  $sKey$  denote the public and secret keys, while  $Enc$  and  $Dec$



**Fig. 2.** a) Layer-wise inference latency in CNN models. b) Time taken for operations with plaintext and ciphertext operands.

represent the encryption and decryption processes. Consider plaintext values  $pt1$  and  $pt2$ . Encrypting  $pt1$  and  $pt2$  using the public key  $pKey$  results in  $ct1 = Encrypt(pt1, pKey)$  and  $ct2 = Enc(pt2, pKey)$ , representing their encrypted forms. A cryptosystem is considered homomorphic concerning a chosen operator (eg: addition, multiplication), denoted as  $\circ$ , if there exists another operator  $\bullet$  such that  $pt1 \circ pt2 = Dec(ct1 \bullet ct2, sKey)$ .

There are multiple homomorphic encryption schemes like Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), Leveled Homomorphic Encryption (LHE) and Fully Homomorphic Encryption (FHE). FHE is the most flexible of the lot allowing for homomorphic operations of addition and multiplication and computations of arbitrary depth using bootstrapping. Several FHE systems have been suggested, such as the BFV, BGV, and CKKS schemes [9]. BFV and BGV allow vector operations on integers, while CKKS facilitates floating-point operations. These schemes enable Single Instruction Multiple Data (SIMD) operations by bundling various values into arrays and converting them into ciphertexts.

While there are existing works that suggest various methods like efficient message packing [16], frequency domain convolution [18], and quantization [20] for addressing the latency of convolution operation in FHE domain, none of them explored using SATs which are more efficient than standard methods for performing convolution.

Our contributions in this work can be summarized as follows:

- We propose to use box filters in combination with SATs to reduce the inference latency of CNNs in FHE environment.
- We extend the filter approximation algorithm previously used for approximating arbitrary 2D filters with a set of box filters to accommodate both 1D and 3D filters.
- Our experiments demonstrate that our proposed approach is faster when compared to standard or frequency domain convolution in FHE without reasonably affecting the classification performance.



## 2 Related Work

Prior works have proposed various approaches for performing convolutions in FHE domain each differing primarily in the way the image pixels or messages are packed into the ciphertexts and the corresponding algorithm for computing convolution. [16] [14] [17] all propose various ways of packing the input image pixels into ciphertext slots and adapting the deep CNNs like Resnet 20/32/44/56/110 to FHE. [27] proposes channel-wise packing, while [13] uses a hybrid packing method that combines multiple existing packing schemes. [20] proposes to use quantization to reduce the inference latency in CNNs and [29] uses binary networks to remove the need for multiplications although significantly affects the accuracy. While all the above-mentioned works use spatial convolution, [18] and [5] put forward the idea of using frequency domain convolution to suit the efficient single instruction multiple data (SIMD) processing approach present in various FHE schemes. In addition to the efficient packing, and convolution algorithm innovations several works have explored acceleration approaches. [1] for the first time explored using GPUs for encrypted inference of CNNs to decrease the latency while [19] proposes high-performance approaches like MPI. [23] proposes a custom accelerator for FHE computations.

Orthogonal to the existing packing schemes, quantization techniques, and algorithms for speeding up convolution in FHE we propose a methodology to improve the convolution latency in FHE using SATs and box filters.

SATs are well-established concepts in computer vision, enabling the rapid computation of the sum of values within any arbitrary subset of a grid, maintaining a constant time complexity [8]. SATs are used to speed up computations for various tasks including texture mapping [4], decomposition of fully connected layers [3], accelerating convolutions with binary [26] and large kernels [28] while [25] generalizes the summed-area tables for n-dimensional inputs. [21] provides an effective algorithm for learning a set of box filters that approximate any arbitrary 2D kernel.

In this work, we use the kernel approximating algorithm to represent any 1D/2D/3D kernel with a set of box filters and SATs to provide an efficient method for convolution in the FHE domain.

## 3 Methodology

### 3.1 Overview

Given an input image  $I$  and filter  $F$  we calculate convolution in the following steps.

- If not encrypted already, every pixel of  $I$  is encrypted into a ciphertext using FHE and SAT of encrypted input is calculated  $I$ .
- Filter  $F$  is approximated with a set of box filters using the algorithm 4.
- SAT and box filters are used to compute the convolution output as illustrated in Figure 4

### 3.2 Encryption

We use HEAAN [6] library which is based on CKKS FHE scheme for our homomorphic encryption and computations. In our packing method each value of the input signal is encrypted into a ciphertext. When multiple values of input are encrypted into a single ciphertext we cant access the individual elements directly and need to go through a considerable overhead to get the element at an index. But with our approach of packing where each value of input is a ciphertext we can avoid that overhead. The proposed methodology is orthogonal to the packing scheme applied to encrypt the input data into ciphertexts. There are innovative packing techniques to reduce the need for individual element access [16] [14] [17] for convolutions but they need to be modified to suit for our approach of using SAT and box filters, we leave this direction for future work.

### 3.3 Approximating Arbitrary Filters with Box Filters

We approximate a kernel with a set of box filters by extending the algorithm proposed by [21] to approximate any 1D/2D/3D filter with a set of box filters. This algorithm outputs a set of box filters which can be used to produce an approximation of any filter as shown in algorithm 1. Each of these box filters are represented by corner points determine box filter position in the original filter space and a scaling factor.

---

**Algorithm 1** Computing filter from set of box filters

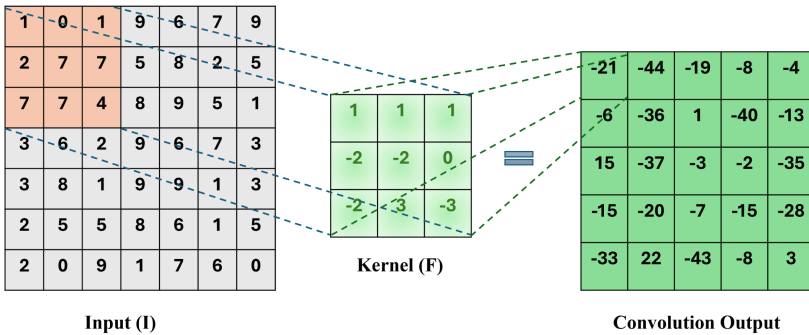
---

```

1: function COMPUTEFILTER( $\{\alpha_i\}_{i=1}^N, \{e_{i1}, e_{i2} \dots e_{ik}\}_{i=1}^N$ )
2:   Initialize filter  $F$  with zeros
3:   for  $i = 1$  to  $N$  do
4:     Compute filter segment  $f_i$  using the edges  $\{e_{i1}, e_{i2} \dots e_{ik}\}$ 
5:      $F = F + \alpha_i \cdot f_i$ 
6:   end for
7:   return  $F$ 
8: end function

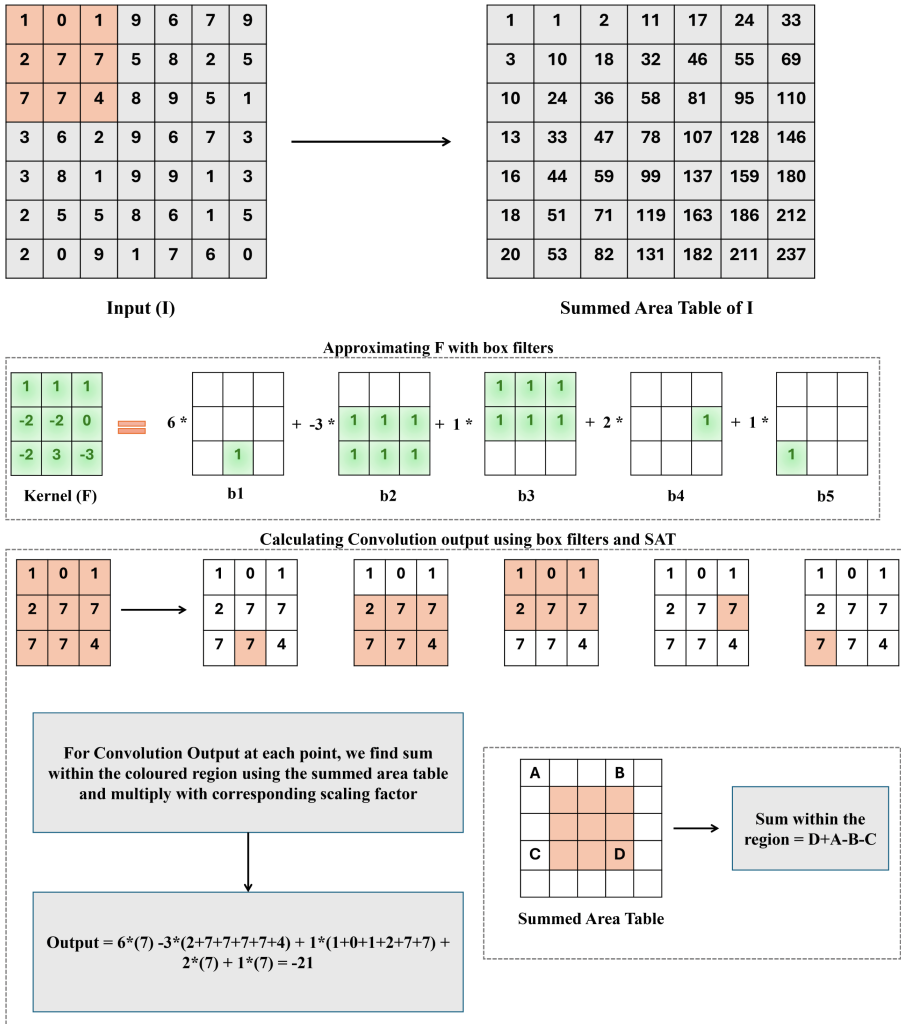
```

---



**Fig. 3.** Illustration of regular spatial convolution

Approximation algorithm 4 is an exhaustive search method which determines the ideal values of the corner points of the box iteratively by working on one box at a time [21]. The approximation algorithm starts from one box and goes iteratively until the maximum number of boxes  $maxN$  is reached or approximation error is below a threshold. For our experiments, we set  $maxN$  based on the filter shape and around 20% to 50% less than the total entries in the filter. Error threshold was set to 1 L2-percent error. At each step of the iteration, optimal



**Fig. 4.** Illustration of proposed approach of convolution with summed-area table and box filters. Given an input image I and kernel F their summed area table and box filter approximations are calculated followed by using them to compute the convolution output

scaling values  $\{\alpha\}$  for the boxes are determined using matrices  $M$  and  $V$ .  $M$  computed using algorithm 2 contains information of overlap between different

---

**Algorithm 2** Compute Overlap Matrix  $M$ 


---

```

1: function COMPUTEM( $n, boxes$ )
2:   Initialize  $n \times n$  matrix  $M$  with all entries as 0
3:   for  $i \leftarrow 1$  to  $n$  do
4:     for  $j \leftarrow 1$  to  $n$  do
5:        $M[i][j] \leftarrow$  Overlap between  $boxes[i]$  and  $boxes[j]$ 
6:     end for
7:   end for
8:   return  $M$ 
9: end function

```

---



---

**Algorithm 3** Compute filter sum Matrix  $V$ 


---

```

1: function COMPUTEV( $n, boxes, filter$ )
2:   Initialize  $n \times 1$  matrix  $V$  with all entries as 0
3:   for  $i \leftarrow 1$  to  $n$  do
4:      $V[i] \leftarrow$  Sum of filter values within the overlap between  $boxes[i]$  and  $filter$ 
5:   end for
6:   return  $V$ 
7: end function

```

---



---

**Algorithm 4** Approximating Filter with Box Filters

---

```

1: function APPROXIMATEFILTER( $maxN, filter, threshold$ )
2:   Start with  $N = 0$ ,  $boxes = []$ ,  $threshold = 1$ 
3:    $k = 2 * Dimension(filter)$ 
4:   while  $N < maxN$  do
5:      $N = N + 1$ .
6:     Keep all values of  $\{e_{i1}, e_{i2} \dots e_{ik}\}_{i=1}^{N-1}$  constant.
7:     for  $e_{n1}$  in range  $(0, e_{n1})$  do ▷ Exhaustive search for corner points
8:        $\vdots$ 
9:     for  $e_{nk}$  in range  $(0, e_{nk})$  do
10:      append the current iteration box  $\{e_{i1}, e_{i2} \dots e_{ik}\}$  to  $boxes$ 
11:       $M = ComputeM(n, boxes)$ 
12:       $V = ComputeV(n, boxes, filter)$ 
13:       $\{\alpha_i\}_{i=1}^N = M^{-1}V$  ▷ Computing scaling factors for current box set
14:       $E(\theta) = \|(ComputeFilter(\{\alpha_i\}_{i=1}^N, \{e_{i1}, e_{i2} \dots e_{ik}\}_{i=1}^N) - filter)\|_2$ 
15:      remove the current iteration box  $\{e_{i1}, e_{i2} \dots e_{ik}\}$  from  $boxes$ 
16:    end for
17:  end for
18:  Add the box  $\{e_{i1}, e_{i2} \dots e_{ik}\}$  to  $boxes$  which had the lowest  $E(\theta)$ .
19:  return  $boxes$ 
20: return  $boxes$ 
21: end function

```

---

boxes in the set, while  $V$  computed using algorithm 3 has information about sum of filter values within the overlap between boxes and the original filter.

### 3.4 Convolution with SAT and Box Filters

Given an  $n$  dimensional input  $I$  and kernel  $F$  we compute the SAT of the input using [26] (for 1D and 2D) or [25] (for 3D) and we obtain a set of  $m$  box filters  $B$  approximating  $F$  through algorithm 4. Equation 1 shows how these set of box filters can be used to get the original kernel  $F$ . As the linear combination of these box filters approximates the original kernel we can apply these box filters individually and combine the responses to get equivalent output of the convolution with the kernel. Since these box filters are rectangles/cuboids of various sizes filled with ones, we just need to get the sum of the values within the rectangle/cuboid and multiply it by the corresponding scaling factor to get the convolution output for a given box filter. SATs come in handy for this process as they provide us with an efficient way to get the sum within a region. The number of multiplications required to calculate convolution output at each point by this approach is equal to the number of box filters used for approximating the kernel, thereby reducing the required multiplications from  $K_1 * K_2 \dots * K_n$  for  $n$  dimension kernel to  $N$  where  $K_1, K_2, \dots, K_n$  are the dimensions of the kernel and  $N$  is the number of boxes required to approximate it.  $n$  dimensional spatial convolution is given by equation 2, using 1 we can replace  $F$  in equation 2 with the set of the box filters  $B$  to obtain equation 3 for computing convolution. Figure 4 shows an example of the approximating box filters, and using SAT to compute convolution.

Our approach also makes the convolution more conducive to parallelization. In addition to parallelization of each channel like in regular spatial convolution we can also parallelize convolution response computation of each box filter effectively reducing the latency further. In all our experiments we use these kinds of parallelization and run parallel threads equal to the number of boxes used.

$$F \approx \alpha_1 b_1 + \alpha_2 b_2 + \dots + \alpha_m b_m \quad (1)$$

$$I * F = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \dots \sum_{k_n=1}^{K_n} I[k_1, k_2, \dots, k_n] \cdot F[i - k_1, j - k_2, \dots, l - k_n] \quad (2)$$

$$\begin{aligned} I * F &= \alpha_1 \sum_{b_{11}=1}^{B_{11}} \sum_{b_{12}=1}^{B_{12}} \dots \sum_{b_{1n}=1}^{B_{1n}} I[b_{11}, b_{12}, \dots, b_{1n}] \cdot b_1 \\ &+ \alpha_2 \sum_{b_{21}=1}^{B_{21}} \sum_{b_{22}=1}^{B_{22}} \dots \sum_{b_{2n}=1}^{B_{2n}} I[b_{21}, b_{22}, \dots, b_{2n}] \cdot b_2 \\ &+ \alpha_m \sum_{b_{m1}=1}^{B_{m1}} \sum_{b_{m2}=1}^{B_{m2}} \dots \sum_{b_{mn}=1}^{B_{mn}} I[b_{m1}, b_{m2}, \dots, b_{mn}] \cdot b_m \end{aligned} \quad (3)$$

where  $B_{i1}, B_{i2}, \dots, B_{in}$  represents the shape of the filter in each dimension for box  $i$ .

## 4 Experiments and Results

We perform various experiments proving the efficiency and efficacy of our approach in computing convolution. These experiments are aimed at understanding the effects of using box filter approximated kernels in terms of error and latency.

### 4.1 Filter Response Error

We perform this experiment to get more insights into the capability of our approach to approximate filters of various types at granular level. We compare individual 1D, 2D, and 3D filters of varying sizes and use the percent error of L2 distance between the filter responses of the original and the approximated filter for this evaluation. We randomly selected kernels learned by standard CNN architectures like Alexnet [15] Resnet [10], VGGNet [24] for 2D, and application specific CNNs [11] [22] found in literature for 1D and 3D. In cases where a specific filter size is not present in the CNN model we used filter with random values. The results are shown in table 1 with average error and average decrease in multiplications over all the kernels tested. The minimum number of boxes required to take the error below 1% over all the kernels tested are shown in Table 1. These results show that we can approximate varied-sized 1D, 2D, and 3D filters with high accuracy while reducing the number of multiplications required for convolution by 22% - 52%.

**Table 1.** Average Filter Response Error with approximate box filters and % of Multiplications reduced for various filters

Dimension	Filter Size	# of Boxes used	% L2 Error in Filter Response	% decrease in Mults
1	7	5	0.94%	28.57%
	14	10	0.87%	28.57%
	28	20	0.49%	28.57%
	56	36	0.94%	35.71%
2	3	5	0.15%	44.44%
	5	20	0.08%	20%
	7	32	0.65%	28.57%
	9	35	0.78%	44.44%
3	3	20	0.74%	25.92%
	4	46	0.95%	28.125%
	5	60	0.87%	52%
	6	130	0.72%	39.81%

## 4.2 CNN Classifier Performance Error

We study the effect of approximating CNN kernels with box filters on the classification performance. We replace all the learned kernels of various CNN architectures with the approximated box filters and report the difference in classification accuracy. We find that the approximation with box filters does not reasonably effect the accuracy of the model as shown in the Table 2.

**Table 2.** CNN Classifier Performance Error when kernels are approximated with box filters

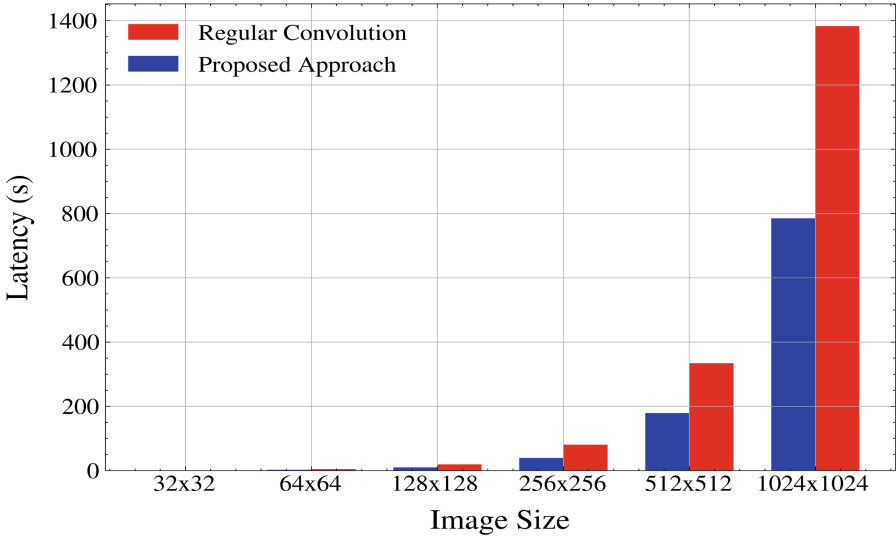
CNN Architecture	Dataset	Accuracy with original kernels (%)	Accuracy when approximated with box filters (%)	% decrease in Mults
ResNet-20	CIFAR-10	91.73	91.73	45.2%
ResNet-18	ImageNet	56.44	56.02	38.32%
ResNet-32	CIFAR-10	92.63	92.59	44.24%
ResNet-34	ImageNet	65.72	64.5	40.17%

## 4.3 Image Size vs Convolution Time

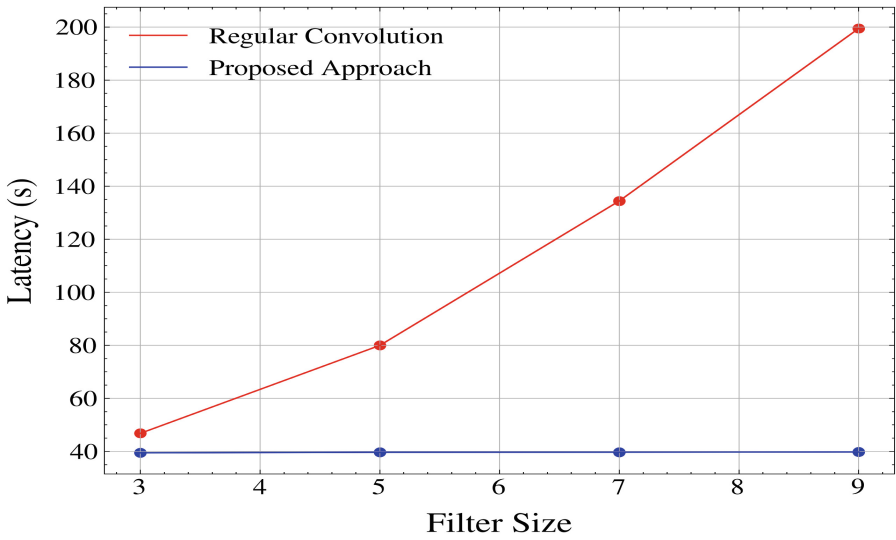
In this experiment, we analyze how the time taken for 2D convolution using our proposed approach scales up with increasing image size in FHE in comparison with regular convolution. For this experiment, we keep the filter size constant at 5x5 and vary the image sizes. Figure 5 shows that the proposed approach is on average twice as fast compared to regular convolution in FHE for image sizes ranging from 32X32 to 1024x1024.

## 4.4 Filter Size vs Convolution Time

We also compare the regular convolution with our proposed method in both 2D and 3D cases with varying filter sizes. For 2D we keep the image size constant at 256x256 and vary the filter sizes from 3 to 9, while for 3D we use an image of shape 32x32x32 and vary the filter sizes from 3 to 6. Based on the results shown in Figure 6, 7 we find that with our approach convolution time remains constant as filter size increases for both 2D and 3D versions while regular convolution shows a quadratic trend.

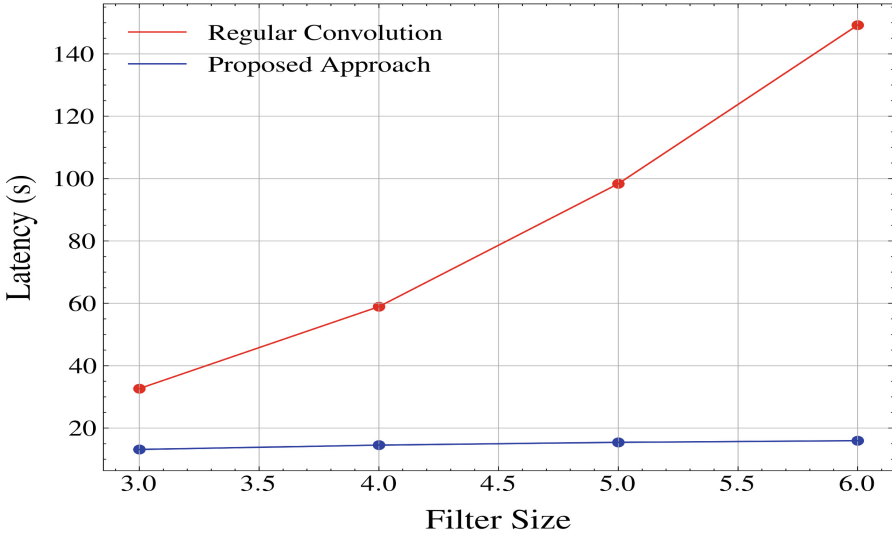


**Fig. 5.** Comparison of latency of proposed approach with regular spatial convolution for varying 2D image sizes with filter size 5x5.

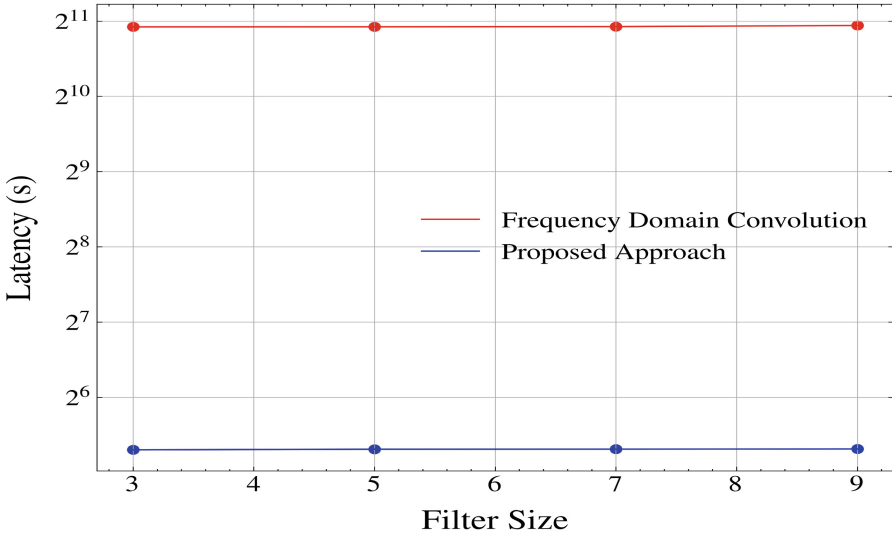


**Fig. 6.** Comparison of latency of proposed approach with regular spatial convolution for varying 2D filter sizes and 256x256 image.





**Fig. 7.** Comparison of latency of proposed approach with regular spatial convolution for varying 3D filter sizes and 32x32x32 image.



**Fig. 8.** Comparison of latency of proposed approach with frequency domain convolution for varying 2D filter sizes and 256x256 image.

#### 4.5 Comparison with Frequency domain Convolution

In addition to comparison of our approach with spatial convolution we also compare it with frequency domain convolution for 2D images and kernels. As

encrypting each pixel is prohibitive for computing Fast Fourier transform (FFT) in FHE we encrypt each row of the input image into a ciphertext and compute FFT of the input. The frequency domain representations of both the input and filter are multiplied and inverse Fourier transform is applied. Inference latency comparison of proposed approach and frequency domain convolution is shown in Figure 8. Although both frequency domain approach and our proposed approach remain constant as filter size increases the former is 64 times slower than our approach.

## 5 Conclusion

In this paper, we proposed an approach to reduce the number of multiplications required in convolution operations using summed-area tables, and box filters. We extend the algorithm proposed in prior work for approximating arbitrary 2D filters with box filters for 1D and 3D versions. We apply our proposed approach for computing convolution in FHE and through various experiments show the efficacy and efficiency of our approach in reducing the convolution latency in the FHE domain. Based on the experimental results we can conclude that our approach proves to be a viable alternative to widely used regular spatial convolution for reducing latency in the FHE domain. A promising line of future work is to explore using our approach in combination with various packing schemes used currently in FHE for inference in CNNs.

## References




1. Al Badawi, A., Jin, C., Lin, J., Mun, C.F., Jie, S.J., Tan, B.H.M., Nan, X., Aung, K.M.M., Chandrasekhar, V.R.: Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. *IEEE Transactions on Emerging Topics in Computing* **9**(3), 1330–1343 (2020)
2. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data* **8**, 1–74 (2021)
3. Babiloni, F., Tanay, T., Deng, J., Maggioni, M., Zafeiriou, S.: Factorized dynamic fully-connected layers for neural networks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 1374–1383 (2023)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mipnerf 360: Unbounded anti-aliased neural radiance fields. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5470–5479 (2022)
5. Bian, S., Wang, T., Hiromoto, M., Shi, Y., Sato, T.: Ense: Efficient secure inference via frequency-domain homomorphic convolution for privacy-preserving visual recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9403–9412 (2020)
6. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) *Advances in Cryptology - ASIACRYPT 2017*, pp. 409–437. Springer International Publishing, Cham (2017)

7. Cong, J., Xiao, B.: Minimizing computation in convolutional neural networks. In: International conference on artificial neural networks. pp. 281–290. Springer (2014)
8. Crow, F.C.: Summed-area tables for texture mapping. In: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. pp. 207–212 (1984)
9. Gorantala, S., Springer, R., Gipson, B.: Unlocking the potential of fully homomorphic encryption. *Commun. ACM* **66**(5), 72–81 (apr 2023). <https://doi.org/10.1145/3572832>, <https://doi.org/10.1145/3572832>
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *CoRR abs/1512.03385* (2015), <http://arxiv.org/abs/1512.03385>
11. John, A., Cardiff, B., John, D.: A 1d-cnn based deep learning technique for sleep apnea detection in iot sensors. *CoRR abs/2105.00528* (2021), <https://arxiv.org/abs/2105.00528>
12. Jung, W., Lee, E., Kim, S., Lee, K., Kim, N., Min, C., Cheon, J.H., Ahn, J.H.: HEAAN demystified: Accelerating fully homomorphic encryption through architecture-centric analysis and optimization. *CoRR abs/2003.04510* (2020), <https://arxiv.org/abs/2003.04510>
13. Kim, D., Park, J., Kim, J., Kim, S., Ahn, J.H.: Hyphen: A hybrid packing method and its optimizations for homomorphic encryption-based neural networks. *IEEE Access* (2023)
14. Kim, D., Guyot, C.: Optimized privacy-preserving cnn inference with fully homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **18**, 2175–2187 (2023)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
16. Lee, E., Lee, J.W., Lee, J., Kim, Y.S., Kim, Y., No, J.S., Choi, W.: Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In: International Conference on Machine Learning. pp. 12403–12422. PMLR (2022)
17. Lee, J.W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.S., et al.: Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access* **10**, 30039–30054 (2022)
18. Lou, Q., Lu, W.j., Hong, C., Jiang, L.: Falcon: Fast spectral inference on encrypted data. *Advances in Neural Information Processing Systems* **33**, 2364–2374 (2020)
19. Meftah, S., Tan, B.H.M., Aung, K.M.M., Yuxiao, L., Jie, L., Veeravalli, B.: Towards high performance homomorphic encryption for inference tasks on cpu: An mpi approach. *Futur. Gener. Comput. Syst.* **134**, 13–21 (2022)
20. Meftah, S., Tan, B.H.M., Mun, C.F., Aung, K.M.M., Veeravalli, B., Chandrasekhar, V.: Doren: toward efficient deep convolutional neural networks with fully homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.* **16**, 3740–3752 (2021)
21. Pires, B.R., Singh, K., Moura, J.M.F.: Approximating image filters with box filters. In: 2011 18th IEEE International Conference on Image Processing. pp. 85–88 (2011). <https://doi.org/10.1109/ICIP.2011.6116693>
22. Rao, C., Liu, Y.: Three-dimensional convolutional neural network (3d-cnn) for heterogeneous material homogenization. *Comput. Mater. Sci.* **184**, 109850 (2020)
23. Samardzic, N., Feldmann, A., Krastev, A., Devadas, S., Dreslinski, R., Peikert, C., Sanchez, D.: F1: A fast and programmable accelerator for fully homomorphic encryption. In: MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture. pp. 238–252 (2021)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)

25. Tapia, E.: A note on the computation of high-dimensional integral images. *Pattern Recogn. Lett.* **32**(2), 197–201 (2011). <https://doi.org/10.1016/j.patrec.2010.10.007>, <https://www.sciencedirect.com/science/article/pii/S0167865510003533>
26. Viola, P., Jones, M.J.: Robust real-time face detection. *Int. J. Comput. Vision* **57**, 137–154 (2004)
27. Xie, T., Yamana, H., Mori, T.: Che: Channel-wise homomorphic encryption for ciphertext inference in convolutional neural network. *IEEE Access* **10**, 107446–107458 (2022)
28. Zhang, L., Halber, M., Rusinkiewicz, S.: Accelerating large-kernel convolution using summed-area tables. arXiv preprint [arXiv:1906.11367](https://arxiv.org/abs/1906.11367) (2019)
29. Zhou, J., Li, J., Panaousis, E., Liang, K.: Deep binarized convolutional neural network inferences over encrypted data. In: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). pp. 160–167. IEEE (2020)



# R-LIME: Rectangular Constraints and Optimization for Local Interpretable Model-agnostic Explanation Methods

Genji Ohara<sup>(✉)</sup>, Keigo Kimura, and Mineichi Kudo

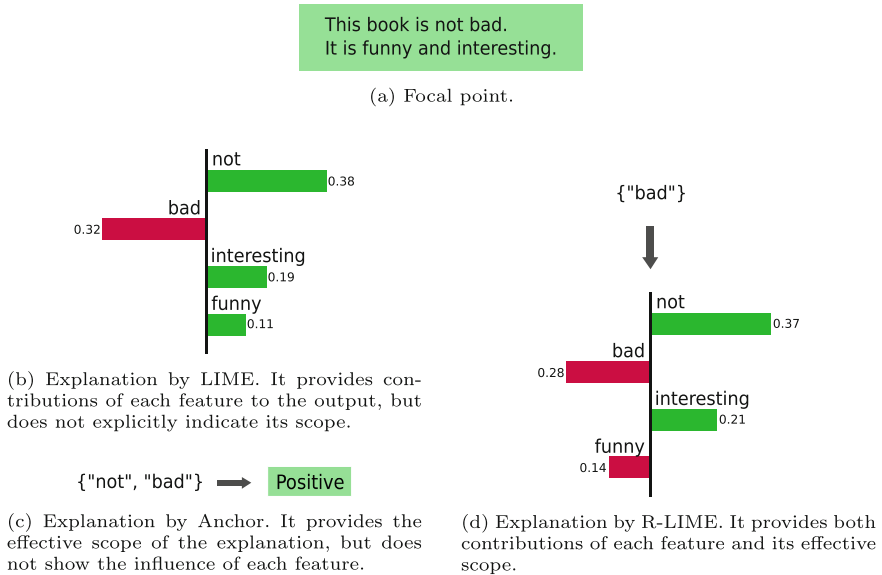
Division of Computer Science and Information Technology, Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan  
{genji-ohara,kimura5,mine}@ist.hokudai.ac.jp

**Abstract.** In recent years, complex machine learning models have been widely introduced in various industrial fields due to their high accuracy. However, their increasing complexity has been a major obstacle to their implementation in sensitive decision-making situations. In order to address this problem, various post-hoc explanation methods have been proposed, but they have not been able to achieve interpretability of both the explanation and its scope. We propose R-LIME, a novel method that interprets complex classifiers within an interpretable scope. R-LIME locally and linearly approximates the complex decision boundary of a black-box classifier within a rectangular region and maximizes the region as long as the approximation accuracy exceeds a given threshold. The resulting rectangular region is interpretable for users because it is expressed as a conjunction of feature predicates. Through qualitative and quantitative comparisons on a real-world dataset, we demonstrate that R-LIME provides more reliable and interpretable explanations than existing methods.

**Keywords:** Interpretable machine learning · Local surrogate model

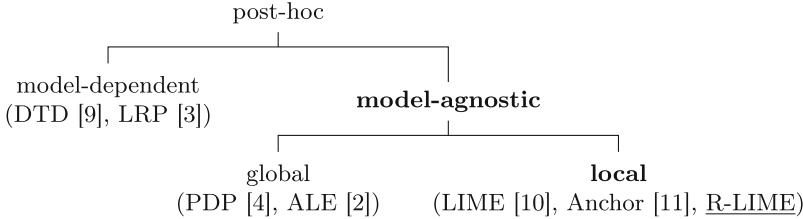
## 1 Introduction

In recent years, complex machine learning models, such as deep neural networks and random forests, have been widely introduced in various industrial fields due to their significant improvements in accuracy. However, their increasing complexity and black-box nature pose challenges, particularly in critical decision-making scenarios such as healthcare and finance, where the lack of transparency becomes a major obstacle to implementation. In order to address this problem, extensive research has focused on post-hoc explanations for complex models [5, 10, 11]. Post-hoc explanation methods are categorized into *model-dependent* and *model-agnostic* methods based on their dependence on the model's structure, with the latter further classified into *global* and *local* methods based on the locality in the input space [13].



**Fig. 1.** Example of explanations by LIME [10], Anchor [11] and R-LIME (our proposed method) for a sentiment prediction model.

In this paper, we focus on *local* and *model-agnostic* methods. LIME [10] and Anchor [11] are representative local model-agnostic methods. An example of explanations by LIME and Anchor for a sentiment prediction model is illustrated in Fig. 1. LIME linearly approximates the complex decision boundary around the given focal point (Fig. 1(a)), then provides the weights of the linear model as the contribution of each feature to the output. The explanation by LIME (Fig. 1(b)) suggests that the word “not” mainly contributes to the positive prediction, but does not explicitly indicate its effective scope. Without the scope, users might mistakenly apply the knowledge derived from the explanation to other instances far from the focal point, potentially leading to misunderstanding of the black-box model’s behavior [11]. For this example, users may apply the derived insights to the sentence “This book is not good.” and mistakenly conclude that the word “not” mainly contributes to the positive prediction for this sentence as well, which is obviously incorrect. Anchor maximizes the coverage of a rectangular region containing the focal point as long as the probability of the black-box classifier outputting the same label as the focal point within the region exceeds a given threshold. While Anchor provides an effective scope of the explanation, users can get less insight compared to LIME. The explanation by Anchor (Fig. 1(c)) suggests that replacing words other than “not” and “bad” has little impact on the classifier’s output. While it clearly cannot be applied to the sentence “This book is not good” because of not including the word “bad”, the explanation does not provide details about the influence of each word, resulting in less user insight into the model’s behavior compared to LIME.



**Fig. 2.** Categorization of post-hoc explanation methods. We focus on *model-agnostic* and *local* methods, which explain model’s local behavior using only its output.

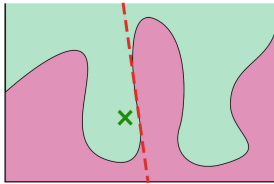
To address these limitations, we propose R-LIME (Ruled LIME), which provides both the contributions of each feature to the output and the effective scope of the explanation. R-LIME linearly approximates a complex decision boundary in a rectangular region and maximizes the region as long as the accuracy of the linear classifier exceeds a given threshold. The region is interpretable for users because it is expressed as a conjunction of feature predicates. An example of the explanation by R-LIME for a sentiment prediction model is shown in Fig. 1(d). It is clear that users can apply the insights derived from the explanation only to the sentences containing the word “bad”.

## 2 Related Work

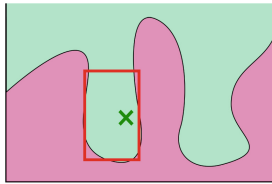
In this section, we overview existing research on post-hoc explanation methods, which explain the behavior of black-box models already trained. As shown in Fig. 2, post-hoc methods are classified into several categories.

They are broadly divided into *model-dependent* and *model-agnostic* methods based on their dependence on the model’s structure. Model-dependent methods, such as deep Taylor decomposition (DTD) [9] and layer-wise relevance propagation (LRP) [3], primarily focus on neural networks and explain the model’s behavior using its parameters [13]. While these methods provide detailed explanations (e.g., layer-wise explanations for neural networks), it is often challenging to apply the same method to models with different structures. In contrast, model-agnostic methods use only the model’s output. Although they are applicable to any model, they cannot explain the internal reasoning processes of the model.

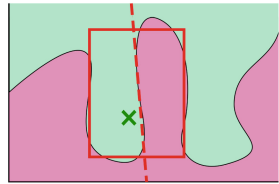
Model-agnostic methods are further categorized into *global* and *local* methods based on their locality in the input space. Global methods, such as partial dependence plots (PDP) [4] and accumulated local effects (ALE) [2], aim to explain the model’s behavior across the entire input space. However, providing global explanations becomes challenging as the model’s complexity increases. In contrast, local methods, such as individual conditional expectation (ICE) [1], local interpretable model-agnostic explanations (LIME) [10], Anchor [11] and shapley additive explanations (SHAP) [8], explain the model’s behavior in the vicinity of a specific input. While they offer explanations simpler and more accurate than global methods, the scope of the explanation is limited locally.



(a) LIME: Locally approximates the decision boundary around the focal point.



(b) Anchor: Maximizes coverage of a rectangular region containing the focal point under accuracy constraints.



(c) R-LIME: Maximizes coverage of a rectangular region under lower constraints on the accuracy of the linear classifier.

**Fig. 3.** Visual comparison of LIME, Anchor and R-LIME (our method). The dashed line represents the local linear approximation model, and the solid line represents the rectangular region containing the focal point.

## 3 Proposed Method

### 3.1 Previous Work

We specifically focus on *local* and *model-agnostic* methods. This section briefly reviews existing research on local model-agnostic explanations, particularly focusing on studies closely related to our proposed method.

**LIME (Local Interpretable Model-agnostic Explanations)** [10] LIME locally approximates a black-box classifier  $f: \mathbb{R}^m \rightarrow \{0, 1\}$  around a focal point  $x \in \mathbb{R}^m$  using a linear classifier  $g: \mathbb{R}^m \rightarrow \{0, 1\}$  (Fig. 3(a)). The approximation process involves the following steps:

1. Generate a set of perturbed samples  $\mathcal{Z}_p$  around  $x$  and their pseudo-labels  $f(\mathcal{Z}_p) = \{f(z) \mid z \in \mathcal{Z}_p\}$ . (i) Convert  $x$  into a binary vector  $x' \in \{0, 1\}^{m'}$ , (ii) generate perturbed samples by randomly drawing non-zero elements from  $x'$ , and (iii) convert the perturbed samples back to the original space.
2. Train a linear classifier  $g$  using  $\mathcal{Z}_p$  and  $f(\mathcal{Z}_p)$  by minimizing the following loss function:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}_p} \pi_x(z) (f(z) - g(z))^2, \quad (1)$$

where  $\pi_x(z)$  is a weight function that assigns larger weights for samples closer to  $x$ , typically defined using an exponential kernel.

LIME provides valuable insights into the local behavior of the model by showing each feature’s contribution to the output  $f(x)$ . However, it does not explicitly define the region for generating perturbed samples, making it difficult for users to assess the effective scope of the explanation [11].

**Anchor** [11] Anchor maximizes the coverage of a rectangular region containing the focal point  $x$ , expressed as a conjunction of feature predicates (a rule) as long as the probability of the black-box classifier  $f$  outputting  $f(x)$  within the region



exceeds a given threshold  $\tau$  (Fig. 3(b)). It aims to highlight important features contributing significantly to the output. For a discrete  $m$ -dimensional input space  $\mathbb{D}^m$ , a trained black-box classifier  $f : \mathbb{D}^m \rightarrow \{0, 1\}$ , an instance  $x \in \mathbb{D}^m$  and a distribution  $\mathcal{D}$  over the input space, a rule  $A(z) = a_{i_1}(z) \wedge a_{i_2}(z) \wedge \dots \wedge a_{i_t}(z)$  is defined. The predicate  $a_i(z)$  evaluates to true ( $= 1$ ) when  $z_i = x_i$  and false ( $= 0$ ) otherwise. The reliability of the explanation is defined as the ‘‘accuracy’’ of the rule, and the generality of the explanation is defined as the ‘‘coverage’’ of the rule. The accuracy  $\text{acc}(A)$  and coverage  $\text{cov}(A)$  of the rule  $A$  are defined as follows:

$$\text{acc}(A) = \mathbb{E}_{z \sim \mathcal{D}(z|A)}[\mathbb{1}_{f(z)=f(x)}], \quad (2)$$

$$\text{cov}(A) = \mathbb{E}_{z \sim \mathcal{D}(z)}[A(z)], \quad (3)$$

where  $\mathcal{D}(z|A)$  is the conditional distribution in the region where the rule  $A$  returns true.  $\text{acc}(A)$  represents the probability that the output of  $f$  matches between the perturbation  $z \sim \mathcal{D}(z|A)$  and the focal point  $x$ , and  $\text{cov}(A)$  expresses the probability that the perturbation  $z$  fits into  $A$ . Anchor maximizes coverage as long as the accuracy of the rule  $A$  exceeds a given threshold  $\tau$ . However, eq. (2) is not directly computable. Introducing a confidence level  $1 - \delta$  ( $0 \leq \delta \leq 1$ ), the accuracy constraint is relaxed as follows:

$$P(\text{acc}(A) \geq \tau) \geq 1 - \delta. \quad (4)$$

Thus, the following optimization problem is solved:

$$A^* = \underset{A \text{ s.t. } P(\text{acc}(A) \geq \tau) \geq 1 - \delta \wedge A(x)=1}{\text{arg max}} \text{cov}(A). \quad (5)$$

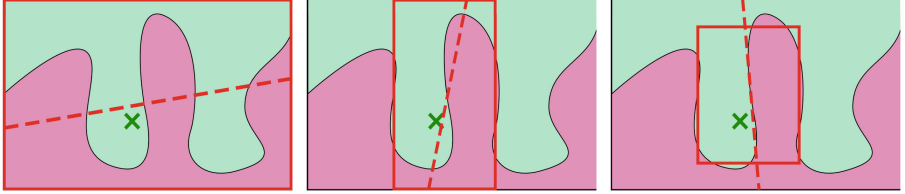
### 3.2 Overview

We propose R-LIME, a novel method designed to address the limitations of LIME [10] and Anchor [11]. Similar to LIME, it locally approximates the black-box classifier  $f$  around the focal point  $x$  using a linear classifier  $g$ , and similar to Anchor, it generates the perturbed samples for approximation from a rectangular region (Fig. 3(c)).

Anchor maximizes the coverage of region  $A$  as long as the probability of the output of the black-box classifier  $f$  matching  $f(x)$  within  $A$  exceeds a given threshold  $\tau$ . R-LIME, on the other hand, learns a linear classifier  $g$  within the rectangular region  $A$  and maximizes the coverage of  $A$  under lower constraints on the accuracy of  $g$ . We modify Anchor’s definition of accuracy in eq. (2) as follows:

$$\text{acc}(A) = \max_{g \in G} \mathbb{E}_{z \sim \mathcal{D}(z|A)}[\mathbb{1}_{f(z)=g(z)}], \quad (6)$$

where  $G$  is a hypothesis space of possible linear classifiers. By solving the optimization problem in eq. (5) under the modified definition of accuracy in eq. (6), we can select the rule that enables explanation with high accuracy and generality.



**Fig. 4.** Overview of the R-LIME algorithm. The progression of the algorithm is illustrated from left to right. The solid line represents the rectangular region  $A$ , and the dashed line represents the linear approximation model  $g$  learned within  $A$ . The initial value of  $A$  is an empty rule (entire input space), and predicates are added to  $A$ , reducing coverage. The process continues until  $\text{acc}(A) \geq \tau$  is satisfied, at which point the rule with the maximum coverage is output.

### 3.3 Algorithm

The R-LIME algorithm is mainly based on the method used in Anchor[11]. For non-convex optimization problems like eq. (5), greedy search are often used. But greedy methods often converge to local optima, so we use beam search, which selects multiple candidates at each iteration to improve the search robustness. The pseudo-code is shown in Algorithm 1.

**Generating New Candidate Rules** To generate new candidate rules, one additional predicate is added to each of the  $B$  candidate rules selected in the previous iteration. The pseudo-code is shown in Algorithm 2.  $T(x)$  is the set of predicates  $\{a_1, \dots, a_m\}$ , where  $a_i(z)$  evaluates to true when  $z_i = x_i$  and false otherwise.  $T(x) \setminus A$  is the set of predicates in  $T(x)$  not included in rule  $A$ .

**Searching Rules with Highest Accuracy** Given the set of candidate rules  $\bar{\mathcal{A}}$ , the algorithm selects the  $B$  candidate rules with the highest accuracy. This problem can be formulated as best arm identification in the multi-armed bandit framework. Each candidate rule  $A_i \in \bar{\mathcal{A}}$  is considered as an arm, and reward of arm  $a_i$  follows a Bernoulli distribution with  $P(X = 1) = \text{acc}(A_i)$ . By sampling  $z \sim \mathcal{D}(\cdot|A_i)$  and obtaining the reward  $\mathbb{1}_{f(z)=g_i(z)}$  for each trial, the algorithm updates  $g_i$  using  $z$  and  $f(z)$  after each trial. To efficiently search the rule (arm) with the highest accuracy, we employ the KL-LUCB algorithm [6]. The pseudo-code is shown in Algorithm 3. For tolerance  $\epsilon \in [0, 1]$ , the KL-LUCB algorithm guarantees below:

$$P(\min_{A \in \bar{\mathcal{A}}} \text{acc}(A) \geq \min_{A' \in \bar{\mathcal{A}}} \text{acc}(A') - \epsilon) \geq 1 - \delta. \quad (7)$$

However, the KL-LUCB algorithm assumes that the reward distribution for each arm remains unchanged, while our method updates the classifier  $g_i$  with each sampling, which may not satisfy the assumption. This issue is discussed further in section 5.2.

---

**Algorithm 1** R-LIME

---

**Input:** Black-box model  $f$ , Target instance  $x$ , Distribution  $\mathcal{D}$ , Threshold  $\tau$ , Beam width  $B$ , Tolerance  $\epsilon$ , Confidence level  $1 - \delta$

**Output:** Rule  $A^*$  satisfying Eq. (5)

```

1:  $A^* \leftarrow \text{null}$ ,  $\mathcal{A}_0 \leftarrow \emptyset$ ,  $t \leftarrow 0$             $\triangleright$  Initialize the set of candidate rules  $\mathcal{A}_0$  to  $\emptyset$ 
2: while  $A^* = \text{null}$  do
3:    $t \leftarrow t + 1$ 
4:    $\bar{\mathcal{A}}_t \leftarrow \text{GENERATECANDS}(\mathcal{A}_{t-1})$ 
5:    $\mathcal{A}_t \leftarrow \text{B-BESTCANDS}(\bar{\mathcal{A}}_t, \mathcal{D}, B, \epsilon, \delta)$ 
6:    $A^* \leftarrow \text{LARGESTCAND}(\mathcal{A}_t, \tau, \delta)$ 
7: end while

```

---



---

**Algorithm 2** Generating new candidate rules

---

```

1: function GENERATECANDS( $\mathcal{A}, x$ )
2:   if  $\mathcal{A} = \emptyset$  then return  $\{true\}$             $\triangleright$  An initial empty rule always returns true
3:    $\bar{\mathcal{A}} \leftarrow \emptyset$ 
4:   for all  $A \in \mathcal{A}$  do
5:     for all  $a \in (T(x) \setminus A)$  do
6:        $\bar{A} \leftarrow \bar{\mathcal{A}} \cup (A \wedge a)$             $\triangleright$  Get a new rule by adding a new predicate  $a$  to  $A$ 
7:     end for
8:   end for
9:   return  $\bar{\mathcal{A}}$ 
10: end function

```

---



---

**Algorithm 3** Searching rules with highest accuracy (KL-LUCB [6])

---

```

1: function B-BESTCANDS( $\bar{\mathcal{A}}, \mathcal{D}, B, \epsilon, \delta$ )
2:   initialize  $\text{acc}, \text{acc}_u, \text{acc}_l$  for  $\forall A \in \bar{\mathcal{A}}$ 
3:    $\mathcal{A} \leftarrow \text{B-PROVISIONALLYBESTCANDS}(\bar{\mathcal{A}})$             $\triangleright$   $B$  rules with highest accuracy
4:    $A \leftarrow \arg \min_{A \in \mathcal{A}} \text{acc}_l(A, \delta)$             $\triangleright$  The rule with the smallest lower bound
5:    $A' \leftarrow \arg \max_{A' \notin (\bar{\mathcal{A}} \setminus \mathcal{A})} \text{acc}_u(A', \delta)$     $\triangleright$  The rule with the largest upper bound
6:   while  $\text{acc}_u(A', \delta) - \text{acc}_l(A, \delta) > \epsilon$  do
7:     sample  $z \sim \mathcal{D}(z|A)$ ,  $z' \sim \mathcal{D}(z'|A')$ 
8:     update  $\text{acc}, \text{acc}_u, \text{acc}_l$  for  $A$  and  $A'$ 
9:      $\mathcal{A} \leftarrow \text{B-PROVISIONALLYBESTCANDS}(\bar{\mathcal{A}})$ 
10:     $A \leftarrow \arg \min_{A \in \mathcal{A}} \text{acc}_l(A, \delta)$ 
11:     $A' \leftarrow \arg \max_{A' \notin (\bar{\mathcal{A}} \setminus \mathcal{A})} \text{acc}_u(A', \delta)$ 
12:  end while
13:  return  $\mathcal{A}$ 
14: end function

```

---



---

**Algorithm 4** Searching a rule with highest coverage under constraint

---

```

1: function LARGESTCAND( $\mathcal{A}, \tau, \delta$ )
2:    $A^* \leftarrow \text{null}$             $\triangleright$  If no rule satisfies the constraint, return null
3:   for all  $A \in \mathcal{A}$  s.t.  $\text{acc}_l(A, \delta) > \tau$  do
4:     if  $\text{cov}(A) > \text{cov}(A^*)$  then  $A^* \leftarrow A$ 
5:   end for
6:   return  $A^*$ 
7: end function

```

---

**Searching a Rule with Highest Coverage under Constraint** To satisfy the constraint imposed by eq. (4), a rule  $A$  needs to meet the following condition:

$$\text{acc}_l(A, \delta) > \tau, \quad (8)$$

where  $\text{acc}_l(A, \delta)$  is the lower limit of the  $100(1 - \delta)\%$  confidence interval for  $\text{acc}(A)$ . If the set of candidate rules  $\mathcal{A}$  includes rules satisfying eq. (8), the one with the maximum coverage among them is selected, then the iteration is terminated. If  $\mathcal{A}$  does not contain any rule satisfying eq. (8), it returns **null**, and proceeds to the next iteration. The pseudo-code is presented in Algorithm 4.

### 3.4 Computational Complexity

Post-hoc explanation methods including LIME, Anchor and R-LIME need to sample a perturbation vector and get the output of the black-box model multiple times, which is computationally expensive. The number of samples required for LIME is  $|\mathcal{Z}_p|$ , which is the number of samples designated by the user. On the other hand, the expected number of samples required for Anchor and R-LIME is bounded by  $\mathcal{O}[m \cdot \mathcal{O}_{\text{MAB}[B \cdot m, B]}]$ , where  $\mathcal{O}_{\text{MAB}[B \cdot m, B]}$  is the expected number of samples for best arm identification finding the best  $B$  arms from  $B \cdot m$  arms. For the KL-LUCB algorithm [6],

$$\mathcal{O}_{\text{MAB}[B \cdot m, B]} = \mathcal{O} \left[ \frac{Bm}{\epsilon^2} \log \frac{Bm}{\epsilon^2 \delta} \right]. \quad (9)$$

Then the total expected number of samples for Anchor and R-LIME is bounded by

$$\mathcal{O} \left[ \frac{Bm^2}{\epsilon^2} \log \frac{Bm}{\epsilon^2 \delta} \right]. \quad (10)$$

For each iteration of the KL-LUCB algorithm, R-LIME needs to update the linear classifier  $g_i$ , which is not required in Anchor. If we use logistic regression as the linear classifier and update it by stochastic gradient descent (SGD) [12], the computational complexity of updating  $g_i$  is  $\mathcal{O}(m)$ . It is negligible compared to the complexity of generating a perturbed sample, which is  $\mathcal{O}(m^2)$  if we get a sample from a multivariate normal distribution using Cholesky decomposition in advance. Overall, the computational complexity of R-LIME is comparable to that of Anchor.

## 4 Experiments

To verify the effectiveness of our method, we conducted qualitative and quantitative comparisons of R-LIME with LIME and Anchor, using a real-world dataset. Our code for R-LIME is available on GitHub<sup>1</sup>.

<sup>1</sup> <https://github.com/g-ohara/rlime>

**Table 1.** Attributes of the recidivism dataset used in the experiments. Continuous features are all discretized, and only binary and ordinal features are considered.

Attribute	Overview	# of Possible Values
Race	Race (Black or White)	2
Alcohol	Presence of serious alcohol issues	2
Junky	Drug usage	2
Supervised Release	Supervised release	2
Married	Marital status	2
Felony	Felony or not	2
WorkRelease	Participation in work release program	2
Crime against Property	Crime against property or not	2
Crime against Person	Crime against a person or not	2
Gender	Gender (Female or Male)	2
Priors	Number of prior offenses	4
YearsSchool	Years of formal education completed	4
PrisonViolations	Number of prison rule violations	3
Age	Age	4
MonthsServed	Months served in prison	4
Recidivism	Recidivism or not	2

#### 4.1 Qualitative Evaluation

**Experimental Setup** We used the recidivism dataset [14] for our experiments. The dataset contains personal information on 9549 prisoners released from North Carolina prisons between July 1, 1979 and June 30, 1980. As shown in Table 1, the dataset includes 19 items such as race (*Race*), gender (*Gender*), presence of alcohol dependence (*Alcohol*), number of prior offenses (*Priors*), and presence of recidivism (*Recidivism*). For this experiment, we treated the binary classification problem of predicting the presence of recidivism (*Recidivism*) as the target label. We discretized continuous features and removed missing values, resulting in 15 features.

We splitted the dataset into training data (7639 instances) and test data (955 instances), and trained a random forest model with 50 trees as the black-box classifier using the training data. Then, we generate LIME, Anchor and R-LIME explanations for two instances extracted from the test data (Fig. 5). For R-LIME, we used logistic regression as the linear approximation model, and a multivariate normal distribution estimated from the training data as the distribution  $\mathcal{D}$ . For both Anchor and R-LIME, the beam width was set to  $B = 10$ , the confidence coefficient to  $1 - \delta = 0.95$ , and the tolerance of the KL-LUCB algorithm to  $\epsilon = 0.05$ . The accuracy threshold  $\tau$  was set to  $\tau = 0.70, 0.90$ .

This problem setting can be considered as a case where a complex machine learning model is introduced to decide parole for prisoners. Since such decisions

Race	Black (0)
Alcohol	No (0)
Junky	No (0)
Supervised Release	Yes (1)
Married	Yes (1)
Felony	No (0)
WorkRelease	Yes (1)
Crime against Property	No (0)
Crime against Person	No (0)
Gender	Male (1)
Priors	1
YearsSchool	$8.00 < \text{YearsSchool} \leq 10.00$ (1)
PrisonViolations	0
Age	$\text{Age} > 33.00$ (3)
MonthsServed	$4.00 < \text{MonthsServed} \leq 9.00$ (1)
Recidivism	No more crimes (0)

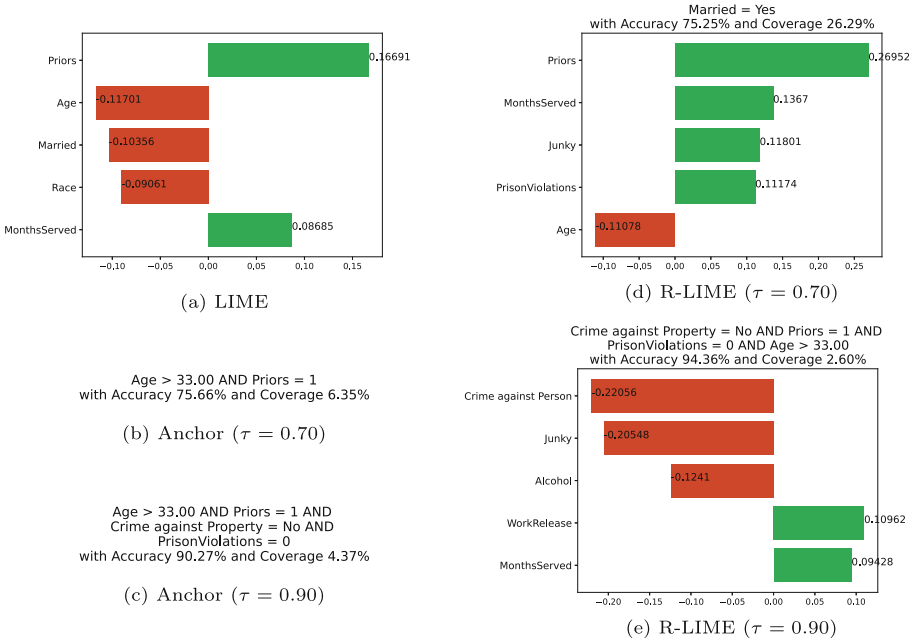
(a) Instance A

Race	Black (0)
Alcohol	Yes (1)
Junky	No (0)
Supervised Release	Yes (1)
Married	No (0)
Felony	No (0)
WorkRelease	Yes (1)
Crime against Property	Yes (1)
Crime against Person	No (0)
Gender	Male (1)
Priors	1
YearsSchool	$\text{YearsSchool} > 11.00$ (3)
PrisonViolations	0
Age	$21.00 < \text{Age} \leq 26.00$ (1)
MonthsServed	$4.00 < \text{MonthsServed} \leq 9.00$ (1)
Recidivism	Re-arrested (1)

(b) Instance B

**Fig. 5.** Two instances sampled from training data of recidivism dataset. Each number in parentheses represents the integer value assigned to the corresponding categorical value.

can have a significant impact on a person’s life, it is crucial for users to appropriately interpret the outputs of black-box models.

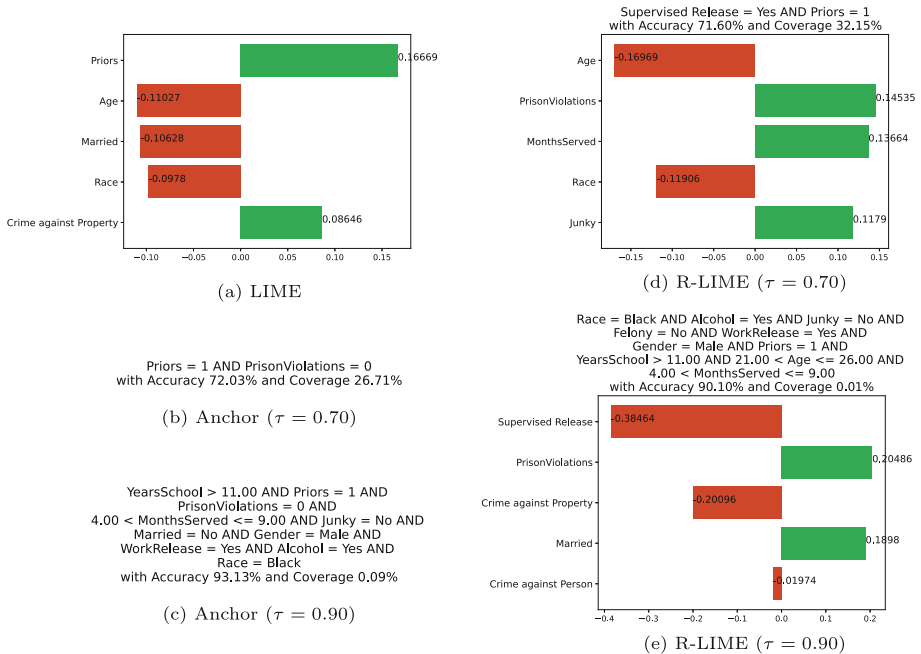


**Fig. 6.** Explanation for Instance A by LIME, Anchor and R-LIME.

**Experimental Results** The results of the experiment are shown in Figs. 6 and 7. The values assigned to each feature name represent the contribution (weight of the linear classifier) to the output of the black-box classifier, normalized such that the absolute sum is 1. The figures display the 5 features with the highest absolute contribution.

Explanations generated by LIME (Figs. 6(a) and 7(a)) provide insights that having a prior offenses (*Priors*), being served for a long time in prison (*Months-Served*), and committing a crime against property (*Crime against Property*) primarily contribute to the positive prediction (prediction that the prisoner will be re-arrested). On the other hand, being elderly (*Age*), being married (*Married*), and being of white race (*Race*) contribute to the negative prediction (prediction that the prisoner will not be re-arrested). While these LIME explanations provide valuable insights into the behavior of the black-box model, they do not explicitly indicate the application scope of the explanations, leaving users unable to determine to which prisoners the explanations are applicable.

Anchor provides conditions for the model’s output to be fixed with high probability. For example, the explanation for instance A under  $\tau = 0.70$  (Fig. 6(b)) means that the model will predict with 75.66 % probability that the prisoner will commit no more crimes when a prisoner is older than 33 and has one prior offense. Although it clearly provides the explanation’s application scope, it does not provide details about how these conditions affect the model’s output.



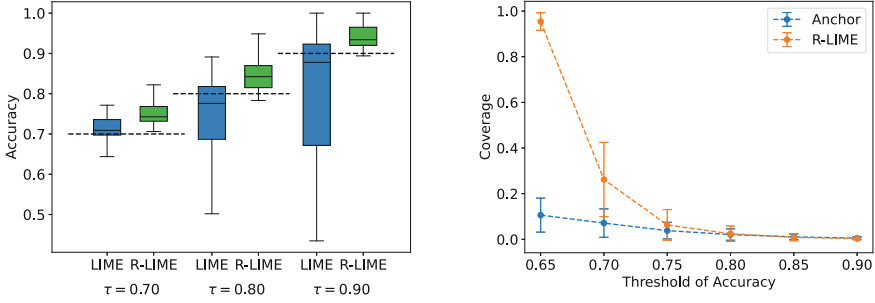
**Fig. 7.** Explanation for Instance B by LIME, Anchor and R-LIME.

In contrast to LIME and Anchor, R-LIME provides both contribution of each feature to the output and the application scope of the explanation. For example, the explanation for instance A under  $\tau = 0.70$  (Fig. 6(d)) indicates that it is applicable only to married prisoner (*Married = Yes*). R-LIME explanations also provide their accuracy and coverage, allowing users to evaluate reliability and generality of the explanations. For example, the coverage of the explanation for instance B under  $\tau = 0.90$  (Fig. 7(e)) is 0.01%, indicating that the decision boundaries around instance B are complex, making it challenging to obtain a high-accuracy linear approximation. This information allows users to discern that the application scope of this explanation is very narrow, limiting its utility.

## 4.2 Quantitative Evaluation: LIME vs. R-LIME

**Experimental Setup** To demonstrate that R-LIME learns a highly accurate linear approximation model in the optimized approximation region, we conducted a comparison of the local accuracy of explanations between LIME and R-LIME. Under the same settings as in section 4.1, we randomly sampled 100 instances from the test data of the recidivism dataset and generated explanations using LIME and R-LIME (with  $\tau = 0.70, 0.80, 0.90$ ). We then sampled 10,000 instances within the rectangular region obtained by R-LIME and calculated the local accuracy of both methods.





(a) Comparison of local accuracy between LIME and R-LIME. R-LIME achieved higher and less variable accuracy compared to LIME.

(b) Comparison of coverage between Anchor and R-LIME. R-LIME achieved higher coverage compared to Anchor for almost values of  $\tau$ , especially for relatively small  $\tau$ .

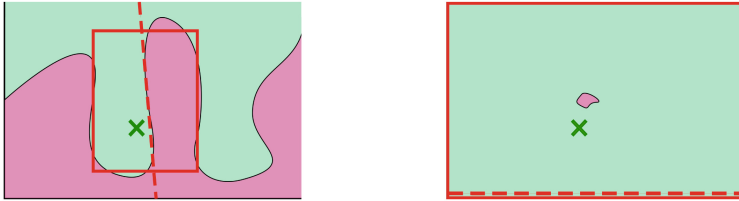
**Fig. 8.** Comparison of existing methods (LIME, Anchor) and R-LIME.

**Experimental Results** The results are presented in Fig. 8(a), showing the distribution of the local accuracy of the linear classifiers learned by LIME and R-LIME. R-LIME achieved higher accuracy compared to LIME for all values of  $\tau$ . This suggests that the linear classifiers learned by LIME and R-LIME differ significantly, and R-LIME learns a high-accuracy linear classifier adapted to the optimized rectangular region. Additionally, as  $\tau$  increases, the variability in the accuracy of LIME widens. This indicates that the linear classifiers learned by LIME may not function effectively as approximation models depending on how the region is selected.

### 4.3 Quantitative Evaluation: Anchor vs. R-LIME

**Experimental Setup** To demonstrate that R-LIME explanations are more general than Anchor, we conducted a comparison of the coverage of explanations between Anchor and R-LIME. Under the same settings as in section 4.1, we generated Anchor and R-LIME explanations for 704 instances from the test data of the recidivism dataset, under the values of  $\tau = 0.65, 0.70, 0.75, 0.80, 0.85, 0.90$ .

**Experimental Results** The results are presented in Fig. 8(b), showing the coverage of the explanations by Anchor and R-LIME. The coverage of explanations generated by R-LIME is higher compared to Anchor for almost values of  $\tau$ , especially for relatively small  $\tau$ . It is because of the flexibility of the linear approximation models learned by R-LIME, which captures the decision boundary more precisely. In contrast, Anchor uses only the intervals of each feature discretized in advance, which cannot capture the decision boundary flexibly and makes its scope narrow.



(a) R-LIME for balanced label distribution. (b) R-LIME for imbalanced label distribution.

**Fig. 9.** Behavior of R-LIME for balanced and imbalanced label distribution. In case of imbalanced label distribution, the approximation region covers the entire input space and the linear approximation model always outputs the majority label.

**Table 2.** Deviation between the estimated accuracy and the true accuracy of the linear classifier learned by R-LIME. The deviation  $0.012 \pm 0.017$  was relatively small considering the confidence level  $1 - \delta = 0.95$ .

	Estimated acc.	True acc.	Deviation
Average	.811	.829	.012
Standard Deviation	.018	.023	.017

## 5 Discussion

### 5.1 Behavior for Imbalanced Label Distribution

R-LIME may generate less useful explanations when the distribution of black-box classifier outputs is imbalanced. When the ratio of outputting the minority label is less than  $1 - \tau$ , where  $\tau$  is the accuracy threshold, the approximation region generated by R-LIME covers the entire input space, and the learned linear classifier always outputs the majority label (Fig. 9).

A first possible solution to this problem is modifying the loss function. Using weighted logistic loss or Focal Loss [7] as the loss function might lead to the generation of more useful explanations in the case of imbalanced label distribution. Another solution involves adding constraints to limit the label distribution bias within the approximation region. In addition to eq. (4), adding a constraint like

$$\left( \mathbb{E}_{z \sim \mathcal{D}(z|A)} [\mathbb{1}_{f(z)=1}] - \frac{1}{2} \right)^2 < \mu \quad (11)$$

could suppress the excessive expansion of the approximation region.

### 5.2 Changes in Reward Distribution in Best Arm Identification

For R-LIME, the problem of selecting the rule with the highest accuracy is formulated as the best arm identification problem in multi-armed bandit framework, and solved using the KL-LUCB algorithm [6]. However, this algorithm

assumes that the reward distribution remains constant, while in R-LIME, the reward distribution (accuracy of the linear approximation) changes with every update of the approximation model after sampling. Therefore, rewards obtained at an early stage might influence the estimated value and make it deviate from the true value.

We conducted an experiment to evaluate the deviation between the estimated accuracy and the true accuracy. We generated explanations for 3200 data instances sampled from the dataset, and compared the estimated accuracy with the true accuracy. The true accuracy was calculated based on 1000 instances sampled within the approximation region. The results in Table 2 show a mean deviation of 0.012 with a standard deviation of 0.017. By considering the confidence level  $1 - \delta = 0.95$ , the deviation was relatively small. While there are concerns about the theoretical validity of using the KL-LUCB algorithm, our results suggest that the deviation is not significant in practice.

### 5.3 Parameter Selection

In Sec. 3.4, we discussed about the computational complexity of R-LIME, which depends on some hyperparameters. R-LIME requires the hyperparameters to be selected by users, such as the threshold of accuracy  $\tau$ , beam width  $B$ , tolerance  $\epsilon$  and confidence level  $\delta$ .  $B$  should be large and  $\epsilon$  and  $\delta$  should be small for accurate results, as long as the computational cost is acceptable. On the other hand,  $\tau$  should be carefully selected by users, sometimes interactively, considering the tradeoff between the accuracy and generality of generated explanation.

## 6 Conclusion

Existing methods for local model-agnostic explanations of black-box classifiers, such as LIME and Anchor, have limitations that they cannot achieve interpretability of both the explanation and its application scope. To address these challenges, we proposed R-LIME, a method that locally and linearly approximates the decision boundary of a black-box classifier and provides a rectangular approximation region, which is interpretable for users due to being expressed as a conjunction of feature predicates. We proposed an algorithm to maximize coverage of the approximation region as long as the accuracy of the linear approximation model exceeds a given threshold. Comparing R-LIME with existing methods on the real-world dataset, we demonstrated that R-LIME achieves interpretability of both the explanation and its application scope, and provides explanations more accurate than LIME and more general than Anchor. Finally, we discussed the instability of behavior with imbalanced label distributions, raised questions about the theoretical validity of using the KL-LUCB algorithm, and hyperparameter tuning in practice.

## References

1. Alex Goldstein, Adam Kapelner, J.B., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>
2. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc. Ser. B Stat Methodol.* **82**(4), 1059–1086 (2020). <https://doi.org/10.1111/rssb.12377>
3. Bach, Sebastian AND Binder, Alexander AND Montavon, Grégoire AND Klauschen, Frederick AND Müller, Klaus-Robert AND Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* **10**(7), 1–46 (2015). <https://doi.org/10.1371/journal.pone.0130140>
4. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **29**(5), 1189–1232 (2001), <http://www.jstor.org/stable/2699986>
5. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems (2018), <https://arxiv.org/abs/1805.10820>
6. Kaufmann, E., Kalyanakrishnan, S.: Information complexity in bandit subset selection. In: Shalev-Shwartz, S., Steinwart, I. (eds.) *Proceedings of the 26th Annual Conference on Learning Theory. Proceedings of Machine Learning Research*, vol. 30, pp. 228–251. PMLR, Princeton, NJ, USA (12–14 Jun 2013), <https://proceedings.mlr.press/v30/Kaufmann13.html>
7. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(2), 318–327 (2020). <https://doi.org/10.1109/TPAMI.2018.2858826>
8. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017)
9. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recogn.* **65**, 211–222 (2017). <https://doi.org/10.1016/j.patcog.2016.11.008>
10. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2939672.2939778>
11. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1), 1527–1535 (Apr 2018). <https://doi.org/10.1609/aaai.v32i1.11491>
12. Robbins, H., Monro, S.: A stochastic approximation method. *The Annals of Mathematical Statistics* **22**(3), 400–407 (1951), <http://www.jstor.org/stable/2236626>
13. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **109**(3), 247–278 (2021). <https://doi.org/10.1109/JPROC.2021.3060483>
14. Schmidt, P., Witte, A.D.: *Predicting Recidivism in North Carolina, 1978 and 1980*. Inter-university Consortium for Political and Social Research (1988)



# Differentially Private Spiking Variational Autoencoder

Srishti Yadav<sup>1</sup>(✉), Anshul Pundhir<sup>2</sup>, Tanish Goyal<sup>3</sup>,  
Balasubramanian Raman<sup>1,2</sup>, and Sanjeev Kumar<sup>1,3</sup>

<sup>1</sup> Mehta Family School of Data Science and Artificial Intelligence,  
Indian Institute of Technology Roorkee, Roorkee, India  
srishti\_y@mfs.iitr.ac.in, bala@cs.iitr.ac.in, sanjeev.kumar@ma.iitr.ac.in

<sup>2</sup> Department of Computer Science and Engineering,  
Indian Institute of Technology Roorkee, Roorkee, India  
anshul\_p@cs.iitr.ac.in

<sup>3</sup> Department of Mathematics, Indian Institute of Technology Roorkee,  
Roorkee, Uttarakhand, India  
tanish\_g@ma.iitr.ac.in

**Abstract.** Spiking Neural Networks (SNNs) are poised to lead the next generation of artificial intelligence, offering energy efficiency and performance on par with traditional neural networks. With these advantages, SNNs are finding widespread applications across various domains. One significant area of interest is image generation using deep learning models like Variational Autoencoders (VAE). However, like other deep learning models, SNNs demand substantial training data to achieve desired outcomes, raising concerns about data privacy. Our pioneering contribution is the introduction of a Differentially Private Spiking Variational Autoencoder (DP-SVAE) for image generation and reconstruction. DP-SVAE employs standard Differentially Private Stochastic Gradient Descent (DP-SGD) to ensure privacy preservation. Additionally, we have evaluated the models against various adversarial attacks to highlight the importance of differential privacy. We comprehensively analyze the proposed model through extensive experimentation across publicly available benchmark datasets. This pioneering study marks the first exploration of privacy considerations in SNN-based VAEs and will catalyze further research in this domain.

**Keywords:** Spiking Neural Networks · Differential Privacy · Variational Autoencoder · Image Reconstruction

## 1 Introduction and Related Work

Artificial Intelligence (AI) has experienced exponential growth across various sectors due to the digitization of industries. Neural networks like Recurrent

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_7](https://doi.org/10.1007/978-3-031-78354-8_7).

Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Artificial Neural Networks (ANNs) have become indispensable components in fields ranging from agriculture, healthcare, social media, robotics, manufacturing, etc. Between 2012 and 2018, the computational resources required for training deep learning models saw a remarkable increase, scaling up by a factor of 300,000, as reported by Schwartz et al.'s work [1]. To overcome the issue of the rapid growth in power consumption to fulfill computational needs, another type of neural network, Spiking Neural Networks (SNNs), is gaining attention due to its energy-efficient behavior. The SNNs are also referred to as the third generation of neural networks [2].

All the aforementioned neural networks share the common goal of emulating the human brain's functionality. Among them, SNNs closely resemble the human brain's operations. To perform computation, SNNs receive input data in the form of spikes at different points in time, which are then processed as per the membrane potential of spiking neurons present in the network. The neuron fires a spike when the membrane potential surpasses a predefined threshold. Various models for spiking neurons are found in the literature, including the Hodgkin-Huxley [3], Leaky-Integrate-and-Fire [4], and Izhikevich neuron models [5]. The main contrast between ANNs and SNNs is their information representation methods. SNNs generally excel in information representation, utilizing differential equations for computation compared to ANNs' activation functions [6]. The inherent spike-firing mechanism of SNNs contributes significantly to their energy efficiency. Despite this energy efficiency, SNNs maintain performance levels comparable to other neural network types [7].

Leveraging their energy efficiency and firm performance, SNNs find applications in various tasks such as speech recognition, image classification, object detection, healthcare, and more [8]. One such important application is image generation. Image generation models need heavy computational resources. Thus, integrating SNNs with such models could offer significant benefits. Skatchkovsky et al. [9] proposed a hybrid Variational Autoencoder (VAE), where the encoder consists of SNN while the decoder uses ANN. Rosenfeld et al. [10] proposed a Spiking Generative Adversarial Network consisting of SNN and ANN. Talafha et al. [11] proposed VAE-sleep based on a biologically realistic sleep algorithm for VAEs. Kamata et al. [12] were the first to propose Fully Spiking VAE (FSVAE) using autoregressive Bernoulli spike sampling, where they achieved better performance on FSVAE compared to its counterpart ANN VAE (built with the same architecture). Moreover, to boost research in domains where data scarcity is present, e.g., medical imaging, SNN-based image generative models have become an efficient alternative for data generation.

In general, creating robust models requires vast data for training and testing; however, storing and utilizing such a large volume of data raises concerns about potential data breaches. Various attacks, such as linkage attacks, membership inference attacks, data reconstruction attacks, adversarial attacks, and model inversion attacks, among others, may compromise the data, model, or output [13]. These privacy threats emphasize the need to develop privacy-preserving techniques for image-generation models.

To address these privacy concerns, Dwork et al. proposed a mathematical framework called Differential Privacy (DP), which can quantify the privacy loss of every data point [14] present in the dataset. DP ensures that if the data is changed by one entry, then the change in the algorithm’s output will be insignificant, i.e., bounded by a small constant value. DP has found applications in almost every domain related to machine learning. Mueller et al. [15] have demonstrated the effects of applying DP on graphical neural networks for graph classification. Xie et al. [16] proposed Differentially Private Generative Adversarial Networks for image generation. Tang et al. [17] explore Differentially Private image classification by learning priors. Wang et al. [18] proposed VideoDP to ensure the privacy of videos using DP. Ziller et al. [19] proposed a DP framework named Deepee, validated on medical imaging tasks. Weggenmann et al. [20] performed text anonymization using DP VAE. Chu et al. [21] proposed DP based denoising diffusion model. Wang et al. [22] were the first to explore DP for SNNs.

Considering the vital importance of DP and the limited exploration of its application in SNNs for image generation using VAEs, we propose differentially private image generation using Spiking VAEs. To the best of our knowledge, we are the first to propose and provide an in-depth analysis of DP image generation using spiking VAEs. Due to its adaptability with most machine learning models, we have followed the standard Differentially Private Stochastic Gradient Descent (DP-SGD) [23] technique to apply DP. We evaluated the privacy-utility trade-off using four benchmark datasets (MNIST [24], FMNIST [25], CIFAR10 [26], and CelebA [27]). To provide a better comparison of SNN and ANN-based models, we implemented DP on both SNN VAE and ANN VAE. We highlight our contributions as follows:

- To the best of our knowledge, we are the first to propose differentially private SNN-VAE (DP-SVAE) for image generation.
- To further estimate the influence of DP on spiking and non-spiking models, we implemented DP on Spiking VAE and ANN VAE models and provided thorough analysis using several quantitative and qualitative measures.
- We rigorously evaluated and demonstrated the robustness of Differential Privacy (DP) by subjecting both differentially private and non-differentially private versions of spiking and non-spiking models to various adversarial attacks.
- We provide a thorough analysis of DP-SVAE using various hyperparameters to evaluate their privacy utility trade-off for image generation.

## 2 Preliminaries

This section provides important background information for differentially private image generation using spiking VAE.

**Variational Autoencoder (VAE):** VAE is commonly used for image generation task [28]. It consists of an encoder, a decoder, and a latent variable

$z$ . For image generation, it trains a latent variable model  $p(x, z)$  defined as  $p(x, z) = p(z) p(x|z)$ , where  $x, p(z)$  denotes the input and probability distribution over latent variable  $z$  (commonly referred as prior distribution) respectively, and  $p(x|z)$  denotes the probability distribution for the decoder. The posterior probability  $p(z|x)$  is intractable, and hence the encoder model,  $q(z|x)$ , was introduced [29]. The variational lower bound on the marginal likelihood of  $p(x)$  can be defined using Eq. 1.

$$\mathcal{L} = -\text{KL}[q(z|x)||p(z)] + \mathbb{E}_{q(z|x)} [\log p(x|z)] \quad (1)$$

Here,  $\text{KL}[q(z|x)||p(z)]$  is called Kullback-Leibler ( $KL$ ) divergence between the posterior and prior.

**Spiking Neural Networks (SNN):** SNNs are considered to be energy-efficient substitutes for other neural networks. SNNs take input features in the form of spikes, which are created using encoding methods, such as direct encoding. The working principle of SNNs is forwarding the spike trains to the next layer when a predefined threshold value of membrane potential is surpassed in the spiking neurons. After firing the spike, the membrane potential resets itself to its resting potential. Some of the most commonly used spiking neuron models include the Leaky-Integrate-and-Fire (LIF) model [4], Hodgkin-Huxley model [3], and Izhikevich model [5]. In this work, the LIF model is used and defined using Eq. 2.

$$\tau_\alpha \frac{du_\alpha(t)}{dt} = -(u_\alpha(t) - u_r) + R \cdot I(t), \quad \text{when } u_\alpha(t) < v_t \quad (2)$$

Here  $\tau_\alpha$  is membrane time constant,  $u_\alpha(t)$  is membrane potential,  $u_r(t)$  is the resting potential,  $I(t)$  is the input current,  $R$  is the resistance, and  $v_t$  is the threshold potential at the time stamp  $t$ .

**Differential Privacy (DP):** DP [14] is a mathematical framework to ensure data privacy. In more general terms, an algorithm is differentially private if the inclusion or exclusion of a single data point does not substantially affect the output. Mathematically, let  $\mathcal{D}_1$  and  $\mathcal{D}_2$  be two neighboring datasets (i.e., both datasets differ by one point). A randomized algorithm  $\mathcal{M}$  is said to be  $(\epsilon, \delta)$  differentially private if for any two input data points  $x \in \mathcal{D}_1, y \in \mathcal{D}_2$ , it follows bound as shown using Eq. 3.

$$\mathcal{P}[\mathcal{M}(x) \in \mathcal{O}] \leq \exp(\epsilon) \mathcal{P}[\mathcal{M}(y) \in \mathcal{O}] + \delta \quad (3)$$

where  $\mathcal{O} \subseteq \text{Range}(\mathcal{M})$  and  $\mathcal{P}$  denotes the probability. DP is generally applied by adding noise to the gradients during model training. Some of the most important noise addition mechanisms include the Laplace mechanism, Gaussian mechanism, Exponential mechanism, etc[30]. In our work, we followed the Gaussian



mechanism to add the noise ( $\mathcal{N}(0, \sigma^2 \Delta^2)$ ) sampled from the Gaussian distribution. Let  $q$  be a query function and  $\Delta$  be  $\mathcal{L}_2$  - Sensitivity of  $q$ , then Gaussian mechanism ( $\mathcal{A}$ ) over dataset ( $\mathcal{D}$ ) can be defined using Eq. 4.

$$\mathcal{A}(\mathcal{D}) = q(\mathcal{D}) + \mathcal{N}(0, \sigma^2 \Delta^2) \quad (4)$$

Here, the Gaussian Mechanism with parameter  $\sigma$  such that  $\sigma \geq \frac{c\Delta(q)}{\epsilon}$  is  $(\epsilon, \delta)$ -differentially private for some constant,  $c \geq \sqrt{2 \ln(\frac{1.25}{\delta})}$  [30].  $\mathcal{L}_2$  - Sensitivity determines the maximum change in the output of two neighboring datasets after applying the query. Let  $q$  be a query function and  $\|\cdot\|_2$  be  $\mathcal{L}_2$  norm over the range of  $q$ , then  $\mathcal{L}_2$  - Sensitivity ( $\Delta$ ) of  $q$  is defined using Eq. 5.

$$\Delta = \max_{d(\mathcal{D}_1, \mathcal{D}_2)=1} \|q(\mathcal{D}_1) - q(\mathcal{D}_2)\|_2 \quad (5)$$

**Differentially Private Stochastic Gradient Descent (DP-SGD):** DP-SGD [23] is an extension to the Stochastic Gradient Descent algorithm and is widely used to train differentially private machine learning or deep learning models. In DP-SGD, the gradients are first clipped according to their  $\mathcal{L}_2$  - sensitivity at each iteration so that no single gradient can make more significant updates with respect to others. After gradient clipping, noise (termed as noise multiplier) sampled from the Gaussian distribution is added to the gradients. One of the most significant properties of DP is composition, which is used to track privacy expenditure during model training.

**Renyi Differential Privacy (RDP):** To measure the privacy spent, we followed Renyi Differential Privacy [31]. Based on Renyi Divergence,  $(\alpha, \epsilon)$ -RDP is considered to be a relaxed version of DP where  $\alpha \in (1, \infty)$ . Let us assume a randomized algorithm  $\mathcal{A}$  that takes  $\mathcal{D}_1$  as its input. The algorithm  $\mathcal{A}$  is said to be  $(\alpha, \epsilon)$ -Renyi DP if for every pair of neighboring datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , if  $\mathcal{D}_\alpha(\mathcal{A}(\mathcal{D}_1) \parallel \mathcal{A}(\mathcal{D}_2)) \leq \epsilon$ , where  $D_\alpha(\cdot \parallel \cdot)$  denotes the Renyi divergence of order  $\alpha$  [32]. Due to its compositional properties, it is used with DP-SGD for privacy accounting.

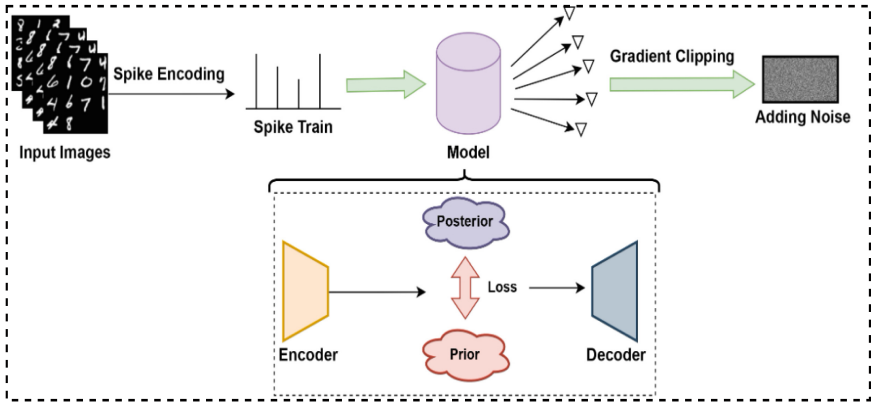
**Adversarial Attacks:** In this study, we employed the following three adversarial attacks during the testing phase to assess the robustness of differentially private and non-differentially private models.

- **Fast Gradient Sign Method (FGSM)** [33] leverages the sign of the gradient from the original input to compute perturbations, thereby generating adversarial examples. These adversarial examples retain perceptual similarity to the original data yet degrade model performance during testing.
- **Carlini and Wagner (C&W) Attack** [34] formulates an optimization problem to generate adversarial examples. The objective function minimizes the perturbation added to the original input while ensuring that the perturbed

input is misclassified. The attack employs advanced optimization techniques to find minimal perturbations that effectively fool the model.

- **Square Attack** [35] generates adversarial images using an iterative search strategy. It effectively finds perturbations by exploring the pixel space of the original image within constraints defined by metrics like the  $l_1$  norm. This method ensures that the perturbations are within a specified distance from the original pixel.

### 3 Differential Private Spiking Variational Autoencoder (DP-SVAE)



**Fig. 1.** The schematic architecture diagram of the proposed approach.

In this work, we develop a differentially private spiking VAE and its equivalent differentially private ANN VAE for image reconstruction and generation. Our approach’s vanilla architecture (i.e., non-differentially private) follows the architecture of FSVAE and ANN VAE as proposed by Kamata et al. [12]. We followed standard DP-SGD training with AdamW optimizer to train our model with DP. We provide the schematic architecture diagram of the proposed model in Fig. 1. The SNNs use binary time series data, so in our proposed DP-SVAE, we used autoregressive Bernoulli spike sampling [12]. The prior and posterior for the latent space of DP-SVAE are defined using Bernoulli distribution [36]. The prior and posterior can be mathematically represented using Eq. 6 and Eq. 7.

$$p(z_{1:T}) = \prod_{t=1}^T p(z_t | z_{<t}) \quad (6)$$

$$q(z_{1:T} | x_{1:T}) = \prod_{t=1}^T q(z_t | x_{\leq t}, z_{<t}) \quad (7)$$

The commonly used loss function for VAE is expressed in Eq.1. For SNN, we use Maximum Mean Discrepancy (*MMD*) [37] alongwith Postsynaptic Potential Function (*PSP*) [38] in place of *KL* divergence.

$$PSP(z_{\leq t}) = (1 - \frac{1}{\tau_{\alpha}})PSP(z_{\leq t-1}) + \frac{1}{\tau_{\alpha}}z_t \quad (8)$$

Here  $\tau_{\alpha}$  denotes the membrane time constant and  $PSP = 0$  when  $z_{\leq 0}$ . The *MMD* metric using *PSP* is defined as,

$$MMD^2 = \sum_{t=1}^T \|PSP(\pi_{q,\leq t}) - PSP(\pi_{p,\leq t})\|^2 \quad (9)$$

Here,  $\pi_{q,t}$  and  $\pi_{p,t}$  represent the average output of the autoregressive SNN model of posterior and prior. They are used as parameters for Bernoulli sampling for latent variables. To start the training of the model, the input image  $I$  is converted into spikes over the time stamps  $T$  using direct encoding and denoted as  $I_{1:T}$ . These spikes are then forwarded to the SNN encoder ( $E$ ) and passed through the *LIF* neurons to obtain the encoded spike trains, denoted as  $I_{1:T}^E$ . This encoded output combined with latent variable  $z_{t-1}$  acts as input to the posterior, which generates  $z_t$  using Bernoulli spike sampling. The prior also has a similar architecture to that of the posterior, but it uses only  $z_{t-1}$  as input. The sampled  $z_t$  is then passed through the decoder to obtain the reconstructed image,  $R_{1:T}$ . The decoder has the same architecture as the encoder. To convert  $R_{1:T}$  into a real image ( $R$ ), non-firing neurons are used in the output layer, which uses membrane potential of  $R_{1:T}$  at the last time stamp  $T$  to obtain the reconstructed image. The overall loss function during training can be described using Eq. 10

$$\mathcal{L} = MSE(I, R) + \sum_{t=1}^T \|PSP(\pi_{q,\leq t}) - PSP(\pi_{p,\leq t})\|^2 \quad (10)$$

To make the model differentially private, we decide the privacy budget,  $\epsilon$ , according to how tight or loose the privacy bound is required. After that, depending on the value of  $\epsilon$ , we train the model with a specific noise multiplier, which determines the Gaussian noise to be added to the model gradients. Along with the noise addition,  $\mathcal{L}_2$ -sensitivity bound is also considered to clip the model gradients. Then, the model is trained using these noisy and clipped gradients to optimize the model. The privacy budget is calculated using *RDP* accountant. We integrated differential privacy into ANN VAE by following the same approach used for DP-SVAE.

## 4 Experimentation and Results

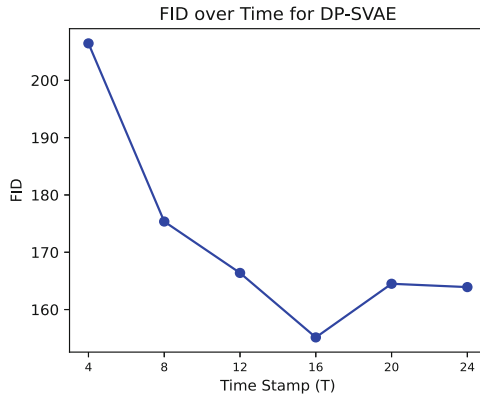
**Dataset Details and Evaluation Metrics:** During experimentation, we used four benchmark datasets, namely MNIST [24], CIFAR10 [26], FMNIST [25],

and CelebA [27]. The MNIST and FMNIST datasets comprise 70,000 grayscale images of handwritten digits and clothing items (60,000 for training and 10,000 for testing). The CIFAR10 dataset consists of 60,000 RGB images (50,000 for training and 10,000 for testing). To further evaluate the scalability of our approach, we have also used the CelebA dataset consisting of 182,732 RGB images (162,770 for training and 19,962 for testing), which contains the celebrity face images. For model evaluation, we used different metrics, i.e., reconstruction loss, Inception Score (IS), and Frechet Distance using Frechet Inception Distance (FID) and Autoencoder Inception Distance (AID). Reconstruction loss quantifies the quality of the reconstructed image with respect to the original image using mean square error. Inception Score evaluates the quality of images generated during image generation. FID examines the statistical likeness between distributions of real and generated images, while IS assesses both the realism and diversity of the generated images. AID assists in computing the Frechet distance of the autoencoder’s latent variables between sampled and real images since datasets such as MNIST are different from the ImageNet domain. Thus, as suggested by Kamata et al. [12], the autoencoder is pretrained on each dataset and utilized for measuring the Frechet distance.

**Implementation Details:** We implemented our approach using PyTorch as a programming framework and Opacus [39] to implement differential privacy for *RDP* accountant. We set  $\delta = \frac{1}{\text{cardinality of the dataset}}$  for RDP accounting. The proposed models were tested on the Ubuntu 22.04 operating system with NVIDIA A5000 GPU and 24 GB graphics memory. We used the official train-test split for evaluation purposes for all the datasets. We followed the same settings as used by [12] and trained the models using AdamW optimizer for 150 epochs with a batch size of 250. For adversarial attacks, we employ perturbations with a magnitude of 0.3 and utilize the mean square error (MSE) as the loss function. In the Square Attack, a perturbation patch size of  $4 \times 4$  is used. We perform 50 iterations to generate adversarial examples for both the C&W and Square attacks.

**Ablation Study:** We have used three different privacy budgets, i.e.,  $\epsilon = \{1, 5, 10\}$  to provide analysis of both loose and tight privacy bounds with respective noise multipliers  $\{0.01, 0.005, 0.001\}$  respectively, which was decided on the fact that stricter privacy bounds require more noise or vice versa. We performed an ablation study using the MNIST dataset on three clip values ( $\{1, 3, 5\}$ ) for all  $\epsilon$  and their respective noise values to determine optimal clip values. We provided the summary of the ablation study on clip values for DP-SVAE and DP-ANN VAE models in Table 1 and Table 2 respectively. From Table 1 and Table 2, we found clip=3 is robust for DP-SVAE and comparable for DP-ANN VAE models for most of the datasets. For the CelebA dataset, we observed that clip=5 is giving better results. Hence, clip value 3 is used throughout the experiments for MNIST, FMIST, and CIFAR10, while clip value 5 is used for the CelebA dataset. During experimentation, we observed that the learning of DP-SVAE and DP-

ANN were sensitive to noise values, hence we have carefully chosen the noise values as  $\{0.01, 0.005, 0.001\}$ . Even on such smaller noise values, we observed a significant performance drop by DP models. We have to increase our privacy budget for these noise values, so we introduced a budget multiplier for privacy accounting. Hence, we multiplied each noise value by 1000 for DP models for privacy budget calculation. We summarize the ablation for the effect of different privacy budgets, i.e.,  $\epsilon = \{1, 5, 10\}$  with their respective noise values on all the datasets for DP-SVAE and DP-ANN VAE in Table 3 and Table 4 respectively. Please refer to the supplementary material for additional experimental results on different clip values. We also analyze the effect of different time stamps,  $T = \{4, 8, 12, 16, 20, 24\}$  on the FID score for DP-SVAE using the MNIST dataset in Fig. 2 and found that  $T = 16$  gives best results.



**Fig. 2.** Ablation study on different time stamps for MNIST dataset using DP-SVAE at  $\epsilon=10$ , noise=0.001, and clip=3.0.

**Table 1.** Ablation study on different clip values =  $\{1, 3, 5\}$  for MNIST dataset using DP-SVAE.

$\epsilon$	Noise	Clip	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
1	0.01	1	0.27	<b>1.159</b>	328.32	413.58
1	0.01	3	0.231	1.089	285.6	<b>357.4</b>
1	0.01	5	<b>0.23</b>	1.066	<b>279.01</b>	386.43
5	0.005	1	0.288	1.109	302.73	428.55
5	0.005	3	<b>0.109</b>	<b>3.844</b>	<b>224.37</b>	<b>169.68</b>
5	0.005	5	0.126	3.635	234.13	180.09
10	0.001	1	0.34	1.031	367.25	838.69
10	0.001	3	0.057	<b>5.288</b>	<b>155.14</b>	<b>55.33</b>
10	0.001	5	<b>0.055</b>	5.115	162.64	57.59

**Table 2.** Ablation study on different clip values = {1, 3, 5} on MNIST dataset using DP-ANN VAE.

$\epsilon$	Noise	Clip	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
1	0.01	1	0.148	4.475	263.19	98.01
1	0.01	3	<b>0.142</b>	<b>4.71</b>	264.34	98.45
1	0.01	5	0.149	4.441	<b>257.82</b>	<b>96.47</b>
5	0.005	1	0.118	<b>5.371</b>	229.55	86.53
5	0.005	3	0.12	5.325	229.01	86.18
5	0.005	5	<b>0.116</b>	5.147	<b>227.96</b>	<b>84.76</b>
10	0.001	1	0.076	5.323	<b>155.81</b>	<b>46.22</b>
10	0.001	3	0.077	5.146	158.71	46.89
10	0.001	5	<b>0.076</b>	<b>5.569</b>	162.4	49.61

**Table 3.** Ablation study of DP-SVAE at different combinations of  $\epsilon$  and noise values for different datasets.

Dataset	$\epsilon$	Noise	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
MNIST	1	0.01	0.231	1.089	285.6	357.4
	5	0.005	0.109	3.844	224.37	169.68
	10	0.001	0.057	5.288	155.14	55.33
CIFAR10	1	0.01	0.249	1.015	405.5	423.48
	5	0.005	0.141	1.169	312.04	191.6
	10	0.001	0.102	1.935	253.06	163.29
FMNIST	1	0.01	0.107	2.274	291.28	146.3
	5	0.005	0.084	2.499	277.74	98.72
	10	0.001	0.061	4.384	210.76	43.95
CelebA	1	0.01	0.265	1.046	406.83	1270.45
	5	0.005	0.102	2.007	337.79	433.78
	10	0.001	0.083	2.67	249.84	291.79

**Table 4.** Ablation study of DP-ANN VAE at different combinations of  $\epsilon$  and noise values for different datasets.

Dataset	$\epsilon$	Noise	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
MNIST	1	0.01	0.142	4.71	264.34	98.45
	5	0.005	0.12	5.325	229.01	86.18
	10	0.001	0.076	5.146	158.71	46.89
CIFAR10	1	0.01	0.159	1.303	261.85	207.66
	5	0.005	0.144	1.536	261.63	208.91
	10	0.001	0.129	1.975	303.33	244.03
FMNIST	1	0.01	0.112	4.286	258.86	66.6
	5	0.005	0.093	4.513	249.43	57.07
	10	0.001	0.072	4.343	216.14	39.76
CelebA	1	0.01	0.115	2.067	318.59	378.63
	5	0.005	0.104	2.154	321.14	341.93
	10	0.001	0.084	2.708	281.57	290.81

**Comparison with State-of-the-Art Methods and Adversarial Attacks:**

In Table 5, we compare our proposed differentially private spiking and non-spiking models against recent state-of-the-art (SOTA) approaches, specifically FSVAE [12] and ESVAE [40]. We observed that non-spiking models experienced a smaller reduction in utility compared to spiking models when subjected to differential privacy. Additionally, although ESVAE, with its complex architecture, initially outperformed FSVAE, its utility degraded more significantly than DP-SVAE under differential privacy constraints. The impact of various adversarial attacks, including FGSM, C&W Attack, and Square Attack, is shown in Table 6, which illustrates an increase in reconstruction loss. To further evaluate the effectiveness of the differentially private models, we quantified the increase in reconstruction loss under adversarial attacks, as shown in Table 7. Our observations indicate that the differentially private models exhibit enhanced robustness against adversarial attacks compared to their non-differentially private counterparts for both spiking and non-spiking variants, demonstrating only a minimal increase in reconstruction loss. Moreover, we also noted that non-spiking models suffer more under adversarial attacks than their spiking models.

Please note that, due to the huge computational demand by large models used for the CelebA dataset, we considered only three datasets for experimentation in Table 5, 6, and 7.

**Table 5.** Analysis using SOTA methods with  $\epsilon=10$ , noise=0.001, and clip=3 for DP (bold entries denote better utility among DP-SVAE and DP-ESVAE).

Dataset	Model	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
<b>MNIST</b>	ANN VAE	0.048	5.947	112.5	17.09
	DP-ANN VAE	0.076	5.146	158.71	46.89
	FSVAE	0.031	6.209	97.06	35.54
	DP-SVAE	<b>0.057</b>	<b>5.288</b>	<b>155.14</b>	<b>55.33</b>
	ESVAE	0.013	5.612	117.8	10.99
	DP-ESVAE	0.073	4.572	235.45	72.81
<b>CIFAR10</b>	ANN VAE	0.105	2.591	229.6	196.9
	DP-ANN VAE	0.129	1.975	303.33	244.03
	FSVAE	0.066	2.945	175.5	133.9
	DP-SVAE	0.102	1.935	253.06	<b>163.29</b>
	ESVAE	0.045	3.758	127.0	14.74
	DP-ESVAE	<b>0.079</b>	<b>2.411</b>	<b>131.33</b>	260.38
<b>FMNIST</b>	ANN VAE	0.05	4.252	123.7	18.08
	DP-ANN VAE	0.072	4.343	216.14	39.76
	FSVAE	0.031	4.551	90.12	15.75
	DP-SVAE	<b>0.061</b>	<b>4.384</b>	<b>210.76</b>	<b>43.95</b>
	ESVAE	0.019	6.227	125.3	11.13
	DP-ESVAE	0.069	3.687	257.781	64.44

**Table 6.** Reconstruction loss during different adversarial attacks on various models (bold entries denote the most severe attack for the given model).

Dataset	Model	No Attack	FGSM	C&W	Square
MNIST	ANN VAE	0.048	0.82191	0.81375	<b>0.97973</b>
	DP-ANN VAE	0.076	0.82139	0.81862	<b>0.99341</b>
	FSVAE	0.031	0.04930	0.04847	<b>0.09353</b>
	DP-SVAE	0.057	0.07734	0.07443	<b>0.08815</b>
	ESVAE	0.013	0.04578	0.04906	<b>0.06310</b>
	DP-ESVAE	0.073	0.08415	0.08560	<b>0.10059</b>
CIFAR10	ANN VAE	0.105	0.23647	<b>0.24365</b>	0.21749
	DP-ANN VAE	0.129	0.21324	<b>0.22775</b>	0.18422
	FSVAE	0.066	<b>0.10641</b>	0.09237	0.10617
	DP-SVAE	0.102	<b>0.13942</b>	0.11110	0.12344
	ESVAE	0.045	0.06735	0.07147	<b>0.09304</b>
	DP-ESVAE	0.079	0.09516	0.09730	<b>0.11118</b>
FMNIST	ANN VAE	0.050	0.61577	0.61633	<b>0.72048</b>
	DP-ANN VAE	0.072	0.62678	0.61790	<b>0.72303</b>
	FSVAE	0.031	0.07710	0.05814	<b>0.08733</b>
	DP-SVAE	0.061	<b>0.11412</b>	0.08307	0.10535
	ESVAE	0.019	0.06301	0.05633	<b>0.09713</b>
	DP-ESVAE	0.069	0.09875	0.08996	<b>0.11898</b>

**Table 7.** Increment in reconstruction loss across various models during different attacks (bold entries indicate models that exhibit the minimum gain in reconstruction loss compared to their non-DP counterparts).

Dataset	Model	FGSM	C&W	Square
MNIST	ANN VAE	0.77391	0.76575	0.93173
	DP-ANN VAE	<b>0.74539</b>	<b>0.74262</b>	<b>0.91741</b>
	FSVAE	0.01830	0.01747	0.06253
	DP-SVAE	0.02034	<b>0.01743</b>	<b>0.03115</b>
	ESVAE	0.03278	0.03606	0.05010
	DP-ESVAE	<b>0.01115</b>	<b>0.01260</b>	<b>0.02759</b>
CIFAR10	ANN VAE	0.13147	0.13865	0.11249
	DP-ANN VAE	<b>0.08424</b>	<b>0.09875</b>	<b>0.05522</b>
	FSVAE	0.04041	0.02637	0.04017
	DP-SVAE	<b>0.03742</b>	<b>0.0091</b>	<b>0.02145</b>
	ESVAE	0.02235	0.02647	0.04804
	DP-ESVAE	<b>0.01616</b>	<b>0.0183</b>	<b>0.03218</b>
FMNIST	ANN VAE	0.56577	0.56633	0.67048
	DP-ANN VAE	<b>0.55478</b>	<b>0.5459</b>	<b>0.65103</b>
	FSVAE	0.04610	0.02714	0.05633
	DP-SVAE	0.05312	<b>0.02207</b>	<b>0.04435</b>
	ESVAE	0.04401	0.03733	0.07813
	DP-ESVAE	<b>0.02975</b>	<b>0.02096</b>	<b>0.04998</b>



## 5 Discussion

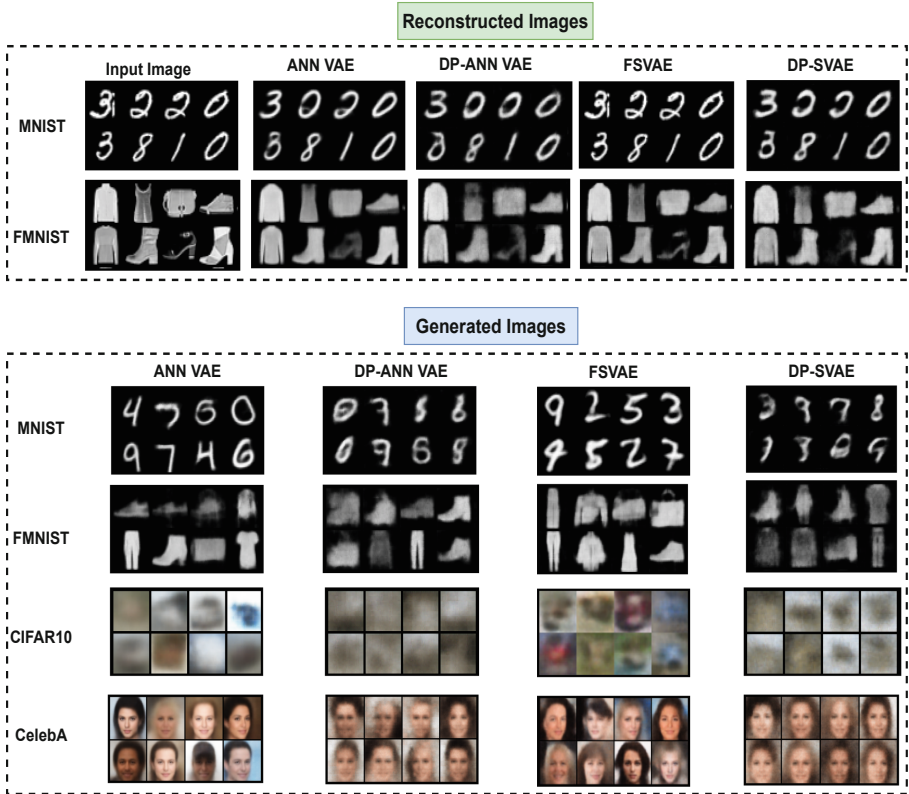
From ablation study (refer to Table 1, Table 2, Table 3, and Table 4), we decided the optimal values for clip,  $\epsilon$ , and noise multiplier. Based on the behavior of different spiking and non-spiking models under DP settings (refer to Table 5, Table 6, and Table 7), we found FSVAE is overall more robust in terms of privacy-utility trade-off. Therefore, we considered FSVAE and its differentially private variant (DP-SVAE) for analysis on all four datasets in Table 8. From Table 3 and Table 4, we observe that as we increase the value of the noise multiplier to get tighter privacy bound, the performance of the DP models degrades. This behavior is not strange but rather a consequence of the privacy utility trade-off. Though all the datasets have shown a similar behavior of degraded performance with an increase in noise, some datasets behave differently in terms of the addition of noise to the gradients and different clip values. Therefore, we have used clip value 3 for MNIST, FMNIST, and CIFAR10, whereas, for CelebA, we used clip value 5 as clip value 3 for CelebA was giving absurd results (please refer to supplementary for the values of CelebA on clip 3). This may be due to the complexity involved in the CelebA dataset in terms of its size and lesser distinguishing features in different celebrity faces compared to other datasets.

**Table 8.** Comparative Analysis of different methods at  $\epsilon=10$ , noise=0.001.

Dataset	Model	Reconstruction Loss ↓	Inception Score ↑	FID ↓	AID ↓
<b>MNIST</b>	ANN VAE	0.048	5.947	112.5	17.09
	DP-ANN VAE	0.076	5.146	158.71	46.89
	FSVAE	0.031	6.209	97.06	35.54
	DP-SVAE	0.057	5.288	155.14	55.33
<b>CIFAR10</b>	ANN VAE	0.105	2.591	229.6	196.9
	DP-ANN VAE	0.129	1.975	303.33	244.03
	FSVAE	0.066	2.945	175.5	133.9
	DP-SVAE	0.102	1.935	253.06	163.29
<b>FMNIST</b>	ANN VAE	0.05	4.252	123.7	18.08
	DP-ANN VAE	0.072	4.343	216.14	39.76
	FSVAE	0.031	4.551	90.12	15.75
	DP-SVAE	0.061	4.384	210.76	43.95
<b>CelebA</b>	ANN VAE	0.059	3.231	92.53	156.9
	DP-ANN VAE	0.104	2.154	321.14	341.93
	FSVAE	0.051	3.697	101.6	112.9
	DP-SVAE	0.083	2.67	249.84	291.79

Also, from Table 8, we can compare the behavior of models when the least amount of noise is added. For example, the Reconstruction Loss, Inception Score, FID, and AID for differentially private and non-differentially private models show similar behavior for MNIST, FMNIST, and CIFAR10 datasets whereas,

for CelebA dataset, we observed significant change in AID and FID by a factor 228.61 and 185.03 for DP-ANN VAE respectively and for DP-SVAE, the FID and AID varies by a factor of 148.24 and 178.89 respectively. From Table 3 and Table 4, we can observe that even the addition of small noise (i.e. 0.01) can cause significant change in reconstruction loss with a maximum value of 0.265 (refer Table 3) in case of CelebA (for DP-SVAE) and 0.159 (refer Table 4) in case of CIFAR10 (for DP-ANN VAE). We have also provided the qualitative comparison of different models (ANN VAE, DP-ANN VAE, FSVAE, and DP-SVAE) in Fig. 3 for image reconstruction and generation, where we used differentially private models with  $\epsilon=10$  and noise = 0.001. From Table 6 and Table 7, we observed that introducing a small noise level (0.001) enhanced the robustness of spiking models against adversarial attacks. In contrast, this noise was insufficient to confer robustness to ANNs, as evidenced by the significant increase in reconstruction loss under various attacks. However, DP models (DP-ANN, DP-SVAE, and DP-ESVAE) demonstrated superior performance compared to their non-DP counterparts under the influence of adversarial attacks (refer to



**Fig. 3.** The reconstructed and generated images obtained by various models for different datasets.

Table 7). In general, we found that after introducing DP, DP-SVAE showed lesser reconstruction loss and a higher Inception Score than DP-ANN VAE. Similarly, DP-SVAE attains a small FID compared to DP-ANN VAE while making them differentially private. Overall, DP-SVAE performed better than DP-ANN VAE, which is consistent with their non-DP variants, but our study also highlights that DP-SVAE is more affected in terms of privacy-utility trade-off comparing DP-ANN VAE.

## 6 Conclusion and Future Scope

In this study, we introduced differentially private implementation of existing SOTA spiking and non-spiking VAE models (DP-SVAE, DP-ESVAE, and DP-ANN VAE), which, to the best of our knowledge, have not been previously proposed. We comprehensively analyzed these models and evaluated their performance using various benchmark datasets. Our study elucidates the privacy-utility trade-off in spiking and non-spiking models. Additionally, we have demonstrated the impact of adversarial attacks, underscoring the potential of differential privacy to enhance model robustness and mitigate the adverse effects of input perturbations. Our results reveal that imposing stricter privacy constraints reduces model utility, with performance variations observed depending on the dataset and model. Notably, we observed a substantial performance decrease in SNNs, even with minimal noise, likely attributed to their spike-driven nature. This underscores the necessity for further research to explore optimized optimization techniques for differentially private SNN models. We also suggest investigating models that offer enhanced privacy and utility for future endeavors. Additionally, including a broader range of real-world datasets could augment the generalizability of our findings.

**Acknowledgement.** We acknowledge the National Supercomputing Mission (NSM) for providing computing resources of “PARAM Ganga” at the Indian Institute of Technology Roorkee, which is implemented by C-DAC and supported by the Ministry of Electronics and Information Technology (MeitY) and Department of Science and Technology (DST), Government of India. We also acknowledge our appreciation for the computational assistance provided by iHub DivyaSampark at IIT Roorkee.

## References

1. Schwartz, R., Dodge, J.: Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM* **63**(12), 54–63 (2020)
2. Maass, W.: Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* **10**(9), 1659–1671 (1997)
3. Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500, 1952
4. Anthony N Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, 95:1–19, 2006

5. Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003
6. Wang, X., Lin, X., Dang, X.: Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Netw.* **125**, 258–280 (2020)
7. Zheng, H., Yujie, W., Deng, L., Yifan, H., Li, G.: Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence* **35**, 11062–11070 (2021)
8. Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural networks*, 111:47–63, 2019
9. Nicolas Skatchkovsky, Osvaldo Simeone, and Hyeryung Jang. Learning to time-decode in spiking neural networks through the information bottleneck. *arXiv preprint arXiv:2106.01177*, 2021
10. Rosenfeld, B., Simeone, O., Rajendran, B.: Spiking generative adversarial networks with a neural network discriminator: Local training, bayesian models, and continual meta-learning. *IEEE Trans. Comput.* **71**(11), 2778–2791 (2022)
11. Sameerah Talafha, Banafsheh Rekabdar, Christos Mousas, and Chinwe Ekenna. Biologically inspired sleep algorithm for variational auto-encoders. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part I 15*, pages 54–67. Springer, 2020
12. Kamata, H., Mukuta, Y., Harada, T.: Fully spiking variational autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence* **36**, 7059–7067 (2022)
13. Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., Lin, Z.: When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)* **54**(2), 1–36 (2021)
14. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4–7, 2006. Proceedings 3*, pages 265–284. Springer, 2006
15. Tamara T Mueller, Johannes C Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, and Georgios Kaissis. Differentially private graph neural networks for whole-graph classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022
16. Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018
17. Xinyu Tang, Ashwinee Panda, Vikash Sehwal, and Prateek Mittal. Differentially private image classification by learning priors from random processes. *Advances in Neural Information Processing Systems*, 36, 2024
18. Han Wang, Shangyu Xie, and Yuan Hong. Videodp: A flexible platform for video analytics with differential privacy. *Proceedings on Privacy Enhancing Technologies*, 2020
19. Ziller, A., Usynin, D., Braren, R., Makowski, M., Rueckert, D., Kaissis, G.: Medical imaging deep learning with differential privacy. *Sci. Rep.* **11**(1), 13524 (2021)
20. Weggenmann, B., Rublack, V., Andrejczuk, M., Mattern, J., Kerschbaum, F.: Dp-vae: Human-readable text anonymization for online reviews with differentially private variational autoencoders. In *Proceedings of the ACM Web Conference* **2022**, 721–731 (2022)
21. Zhiguang Chu, Jingsha He, Dongdong Peng, Xing Zhang, and Nafei Zhu. Differentially private denoise diffusion probability models. *IEEE Access*, 2023

22. Jihang Wang, Dongcheng Zhao, Guobin Shen, Q Zhang, and Y Zeng. Dpsmn: a differentially private spiking neural network. *arXiv preprint arXiv:2205.12718*, 1, 2022
23. Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016
24. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**(6), 141–142 (2012)
25. Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017
26. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009
27. Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015
28. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013
29. Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019
30. Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014
31. Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019
32. Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017
33. Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014
34. Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. Ieee, 2017
35. Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020
36. Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015
37. Diego Arribas, Yuan Zhao, and Il Memming Park. Rescuing neural spike train models from bad mle. *Advances in Neural Information Processing Systems*, 33:2293–2303, 2020
38. Zenke, F., Ganguli, S.: Superspike: Supervised learning in multilayer spiking neural networks. *Neural Comput.* **30**(6), 1514–1541 (2018)
39. Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021
40. Qiugang Zhan, Xiurui Xie, Guisong Liu, and Malu Zhang. Esvae: An efficient spiking variational autoencoder with reparameterizable poisson spiking sampling. *arXiv preprint arXiv:2310.14839*, 2023



# Balancing the Scales: Enhancing Fairness in Facial Emotion Recognition with Latent Alignment

Syed Sameen Ahmad Rizvi<sup>(✉)</sup>, Aryan Seth<sup>(ID)</sup>, and Pratik Narang<sup>(ID)</sup>

Birla Institute of Technology and Science, Pilani Campus, Pilani, India  
{p20190412,f20212221,pratik.narang}@pilani.bits-pilani.ac.in

**Abstract.** Automatically recognizing emotional intent using facial expression has been a thoroughly investigated topic in the realm of computer vision. Facial Expression Recognition (FER), being a supervised learning task, relies heavily on substantially large data exemplifying various socio-cultural demographic attributes. Over the past decade, several real-world in-the-wild FER datasets that have been proposed were collected through crowd-sourcing or web-scraping. However, most of these practically used datasets employ a manual annotation methodology for labelling emotional intent, which inherently propagates individual demographic biases. Moreover, these datasets also lack an equitable representation of various socio-cultural demographic groups, thereby inducing a class imbalance. Bias analysis and its mitigation have been investigated across multiple domains and problem settings; however, in the FER domain, this is a relatively lesser explored area. This work leverages representation learning based on latent spaces to mitigate bias in facial expression recognition systems, thereby enhancing a deep learning model's fairness and overall accuracy.

**Keywords:** Bias Mitigation · Facial Expression Recognition · Fairness

## 1 Introduction

Facial emotion recognition (FER) has been a deeply explored problem in the field of machine learning and computer vision. In the past decade, numerous proposed FER datasets have made it easier to approach facial expression recognition as a supervised deep-learning task. Deep learning requires large and diverse datasets for efficaciously modelling data distribution. However, such a supervised learning strategy necessitates substantial training data that reflects a wide range of socio-cultural demographic characteristics.

Over the past decade, various real-world, in-the-wild datasets have been proposed using web-scraped/crowd-sourced images. A crucial drawback of employing such a data-driven method for expression recognition lies in its susceptibility to biases present in the datasets, particularly those that disproportionately

Supported by Kwikpic AI Solutions.

S. S. A. Rizvi and A. Seth—Equal contribution

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15315, pp. 113–128, 2025.  
[https://doi.org/10.1007/978-3-031-78354-8\\_8](https://doi.org/10.1007/978-3-031-78354-8_8)

affect specific demographic groups.[3,11]. Facial Emotion Recognition requires human annotations for each image, which propagates the biases and prejudices of the annotators. Moreover, most real-world in-the-wild datasets lack proportionate representation of different demographic attributes such as race, age, and gender. Another crucial factor contributing to bias in FER datasets is crowd-sourced annotation. Each annotator possesses their own bias with respect to understanding facial expressions in varied demographics. However, given the enormous size of datasets, these biases are often assumed to be components of random noise.[2,47].

In practice, however, people may harbour systematic and demographic biases, especially when inadequately trained with proper demographic and psychological knowledge; they may incorporate such biases into their annotations [6]. Bias is defined as systematic mistakes that result in unjust outcomes during a decision-making process. Within deep learning, this can originate from multiple factors, such as data collection methodology, algorithm design, and biased human annotation [7]. A deep learning model trained on such datasets would inherently propagate bias, thus making it unfair. Fairness in the context of deep learning refers to the absence of bias or discrimination in deep learning systems; however, achieving it can be difficult since deploying a real-world deep learning solution can propagate biases that can emerge in such systems.

Annotation biases and imbalances in class distribution and demographic representation within datasets amplify biases and undermine equal-odds fairness across attributes like gender and ethnicity. This underscores the importance of scrutinizing dataset bias and developing algorithms to mitigate its effects. When examining age as a specific attribute, it becomes evident that younger individuals are more often depicted with positive emotions (e.g., happiness) [6], whereas older adults are more frequently associated with negative emotions (e.g., sadness and disgust). This reveals a bias in the models, which tend to perceive younger individuals more positively, while older adults are more likely to be assigned negative emotional predictions.

Bias analysis and its mitigation strategies have gained good traction among researchers working in the facial analysis domain. However, in the FER domain, this is a relatively less explored area [34,42]. This research work seeks to tackle and diminish this bias, aiming to enhance fairness in deep learning models. The key contributions of this research encompass:

- A novel latent alignment technique with an architecture that creates better latent representations, mitigates bias, and increases accuracy for Facial Emotion Recognition.
- A novel training technique and loss function that uses Variational Autoencoders and an adversarial discriminator with perceptual loss for bias mitigation and a CNN backbone for expression classification.
- Conducting extensive evaluation on two popular datasets (RAF-DB [26] and CelebA[28]) and multiple protected attributes in both separate and combined techniques, mitigating bias towards gender, race, and age, setting new state-of-the-art results and competitive performance.

This paper is an expanded version of our Student Abstract published at AAAI-24 [35], which, as far as we know, represents the first effort to explore the use of latent space representation learning for mitigating biases in the facial expression recognition domain. This paper provides more comprehensive experimentation with an additional dataset (CelebA[28]), detailed results on the interplay between different protected attributes, and better insights into our methodology and training approach.

The remainder of this paper is structured as follows: Section 2 reviews recent significant contributions in bias mitigation. Section 3 outlines the adopted methodology, detailing the training process, loss functions, and the classification model used. Section 4 showcases our experimental results, including the evaluation metrics and dataset analysis. Section 5 offers a component-wise ablation study of the proposed architecture. Finally, Section 6 concludes the study and suggests directions for future research.

## 2 Recent Works

Bias in Machine learning has attracted wider attention in recent years, with the rapid growth in the deployment of real-world machine learning applications. Extensive surveys[9, 17, 29, 32] have been done to study bias and its mitigation strategies. In this section, we discuss some of the notable methods for mitigating biases. The literature [9] identifies three primary strategies for mitigating bias, categorized as pre-processing, in-processing, and post-processing techniques.

*Pre-processing techniques:* In [4] an optimized pre-processing strategy was presented that modifies the data features and labels. Zemel et al. [43] proposed a strategy for bias mitigation that involves learning fair data representations by framing fairness as an optimization problem, where the goal is to find representations that accurately reflect the data while obfuscating any information regarding membership in protected groups. Feldman et al. [14] proposed disparate impact remover, where feature values were modified while preserving rank ordering to improve overall fairness.

*In-processing:* Kamishima et al. proposed a prejudice remover mechanism [23] that leverages a discrimination-aware regularization approach to the learning objective that can be applied to any prediction algorithm with probabilistic discriminative models. Zhang et al. [45] introduced a strategy for learning fair representations by incorporating a variable representing the group of interest while simultaneously training both a predictor and an adversary. The Meta Fair Classifier [5] suggests a meta-algorithm for classification that integrates fairness constraints into its input and produces an optimized classifier as output.

*Post-processing:* Reject option Classification [22] presents a discriminative aware classification, which essentially aims at the prediction that carries a higher degree of uncertainty and consequently allocates positive outcomes to underrepresented groups and negative outcomes to more advantaged groups. The calibrated equalized odds strategy [33] aims to optimize the output scores of a calibrated classifier



by adjusting the probabilities to modify output labels, all while upholding the goal of equalized odds.

Some other techniques to tackle *dataset bias* include transfer learning[31], adversarial mitigation[39, 46], and domain adaptation [36–38]. Various strategies have been proposed to eliminate or prevent models from acquiring misleading or unwanted correlations. A post-hoc correction technique [15] that imposes an equality of odds constraint on previously learnt predictor. In the domain of deep learning, two popular techniques are the tweaking of loss functions to impose penalties on unfairness[1], and adversarial learning [20, 30, 45]. These techniques aim to learn a fair representation that is devoid of any information related to protected attributes.

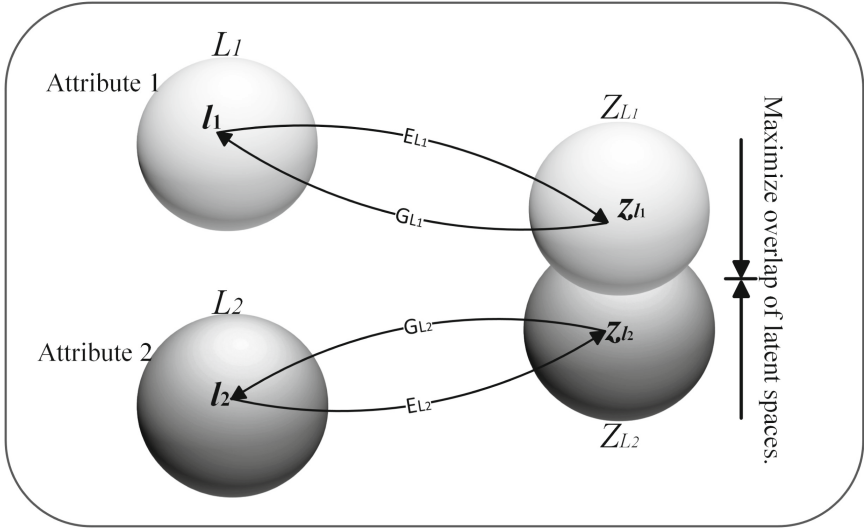
**Bias mitigation in Facial Affect Recognition:** Bias mitigation in affect recognition is a relatively less-explored area. With the exponential increase in computing capabilities over the past decade, many datasets and algorithms have been proposed for automatically recognizing facial expressions. However, most of these in the wild real-world datasets are either web-scraped or crowd-sourced. These datasets often have two major limitations [25]. Firstly, most datasets have class imbalances; i.e. people with varied socio-cultural-ethnic identities are inadequately represented among various classes. Secondly, since these large numbers of scraped images are manually labelled by a group of annotators, a personal bias is inherently a part of the dataset.

Some of the existing works that have tackled bias and it’s mitigation in affect recognition include a facial action unit calibrated FER approach [8], an attribute aware and a disentangled method [42]. Zeng et al. [44] proposed a Meta-Face2Exp architecture that utilized large unlabelled facial recognition datasets.

### 3 Methodology

We propose a two-part model for mitigating bias. Recognizing that CNNs tend to learn from all input features, for the first part of the model we propose a Variational Autoencoder (VAE) to encode the images into a latent space. The images corresponding to each protected attribute in the dataset will each have a corresponding latent space. Our goal is to minimize the distance between these latent spaces so that each latent encodes only the information relevant to expression classification. We propose to utilize a Variational Autoencoder with shared weights for all protected attributes where the inter-latent domain gap is reduced using an adversarial discriminator. We denote the Encoder part as E and the Generator part as G. We introduce a two-part model to address bias mitigation. We develop a two-phase model to address the mitigation of bias. Given the propensity of CNN models to assimilate all input features, the first part of our approach employs a Variational Autoencoder (VAE) which encodes all images, each with a corresponding protected attribute, into the common latent space. The goal is to minimise disparities between these latent spaces, ensuring they contain information relevant to expression classification.

Summarising the methodology:



**Fig. 1.** Framework of Attribute Disentanglement:  $L_i$  denotes the data associated with attribute  $q_i$ .  $Z_{L_i}$  represents the latent space corresponding to  $L_i$ .  $E_{L_i}$  is a variational autoencoder (VAE) with weights shared across  $\forall i$ . 'E' refers to the Encoder module which compresses the input image into a latent which does not contain information about the protected attribute. 'G' refers to the Generator, which is a reconstruction module that converts the latent back to the original image.

- The main cause of bias is that models tend to learn protected attributes as features.
- Our model solves this by generating a latent that has forgotten the protected attribute.
- This is done by overlapping the latent spaces of data points belonging to different protected attributes; this overlap is done using the discriminator.

**Attribute Disentanglement** - We use a Variational Autoencoder with shared weights across the designated protected attributes for that dataset, mitigating domain disparities between latents through an adversarial discriminator. The Encoder and Generator components are represented as 'E' and 'G' as demonstrated in Fig. 1, where  $q_i$  represents protected attributes  $i$ , such as gender.

$$\begin{aligned} \mathcal{L}_{\text{VAE}}(x) = & \text{KL}(z_x | x) \|\mathcal{N}(0, I) + \mathcal{L}_{\text{VAE,D}}^{\text{Latent}}(x) \\ & + \alpha \left\| G_j^\phi(\hat{y}) - G_j^\phi(y) \right\|_F^2 \end{aligned} \quad (1)$$

Equation 1 denotes the objective function for the VAE. The first part of the objective is KL-divergence penalizing the deviation of latent distributions from a Gaussian Distribution. The second part of the objective is a discriminator loss, which determines whether the discriminator correctly predicts the class of

the protected attribute. The final part of the objective is a Style-Reconstruction Loss [21].

**Classification Model** We pass the latent representation generated by E into a classification module consisting of MBCConv[18] blocks demonstrated in Fig. 2.

$$\min_{E, x_i, G, x_i} \max_{D, x_i} = \mathcal{L}_{VAE}(x) + \mathcal{L}_{VAE, D}^{\text{latent}}(x_{q_i}) \quad \forall q \quad (2)$$

**Training Method** The Encoder and Discriminator modules are trained together using a min-max objective function (Equation 2), where the Discriminator employs a categorical cross-entropy loss. Following the VAE training, the classification module is trained separately, utilizing a symmetric cross-entropy loss to enhance robustness.

**Training Configuration** The training was performed on two NVIDIA Tesla V100 GPUs, each with 32 GB of memory. A Stochastic Gradient Descent Optimizer was utilized, configured with a learning rate of 0.0001 and a momentum of 0.9. The hyper-parameter  $\alpha$  in  $L_{VAE}$  from Equation 1 in the paper was determined to be 10 following a grid search.

RAF-DB [26] provides images resized to 128x128 pixels. We applied basic augmentations to our dataset, including horizontal flips with a probability of 50% and random rotations by a maximum angle of 15°.

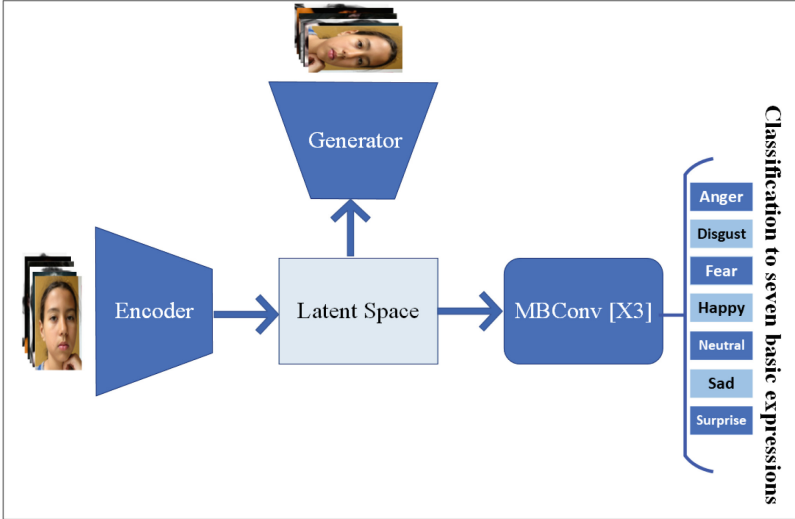
**Loss Functions** The proposed model introduces a novel loss function (Equation 1) that is composed of three distinct components.

The first component is the KL Divergence between the latent variables and a sample from a Gaussian distribution with a mean of 0 and a variance of 1, as described by [24]. This element is employed to create denser representations in the latent space, which enhances accuracy and reduces bias, as demonstrated later in Section 5.

The second component is the loss from the discriminator’s potential to predict the protected attribute accurately. The Encoder’s goal is to be able to fool the discriminator into not knowing the protected attribute. This is the main component that aligns the latent spaces and ensures the Encoder does not learn the protected attribute features.

The final component is the Style-Reconstruction Loss from [21], which is added to ensure that the semantic emotion-level features are not lost on the Generator’s reconstruction of the image. This is used instead of a pixel-wise loss because expression is a subjective concept, and a pixel-wise loss does not necessarily represent semantic consistency.

$$G_j^\phi(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \quad (3)$$



**Fig. 2.** The classification backbone utilizes the latent representation produced by the encoder to categorize the data into seven distinct emotions.

Equation 3 represents the Gram matrix of the  $j_{th}$  feature map for a network  $\phi$ , where  $\phi_j(x)$  corresponds to the activations of the  $j_{th}$  layer in the network. The overall loss is computed as the squared Frobenius norm between the input and output feature matrices.

**Classification Model:** We have used 3 sequential MBCConv [19] modules which use the latent representation generated by the Latent Alignment VAE and classify it into the seven basic expressions. The MBCConv block has been extensively explored in many areas of deep learning and is a versatile and efficient building block. We have also experimented with using Residual Blocks [16] and found that they have a minor reduction in accuracy (described further in Section 5).

## 4 Expermination, Results, and Analysis

### 4.1 Evaluation Metric

We formulate our metric for fairness as [42] and use the “equal odds” philosophy.

$$\mathcal{F} = \min\left(\frac{\sum_{c=1}^C p(\hat{y} = c \mid y = c, q = q_i, \mathbf{x})}{\sum_{c=1}^C p(\hat{y} = c \mid y = c, q = d, \mathbf{x})}\right). \quad (4)$$

$$\forall i \in (1, 2, \dots, N)$$

In equation 4, " $d$ " represents the protected attribute with the highest accuracy. To measure fairness, we compute the accuracy for each class across all attributes and use the minimum value as our fairness metric. For comprehensive analysis, we also calculate the average accuracy for each class across all attributes, following the methodology described in [40].

## 4.2 Experiments and Analysis

We conducted experiments on the RAF-DB [26] and CelebA [28] datasets, following a methodology similar to that in [42]. The RAF-DB dataset comprises 7 classes annotated by humans. Our model utilizes the provided train-test split, with 12,271 images for training and 3,068 images for testing. As shown in Tables 1 and 5, our model achieves state-of-the-art performance on RAF-DB for both metrics, demonstrating significant bias mitigation.

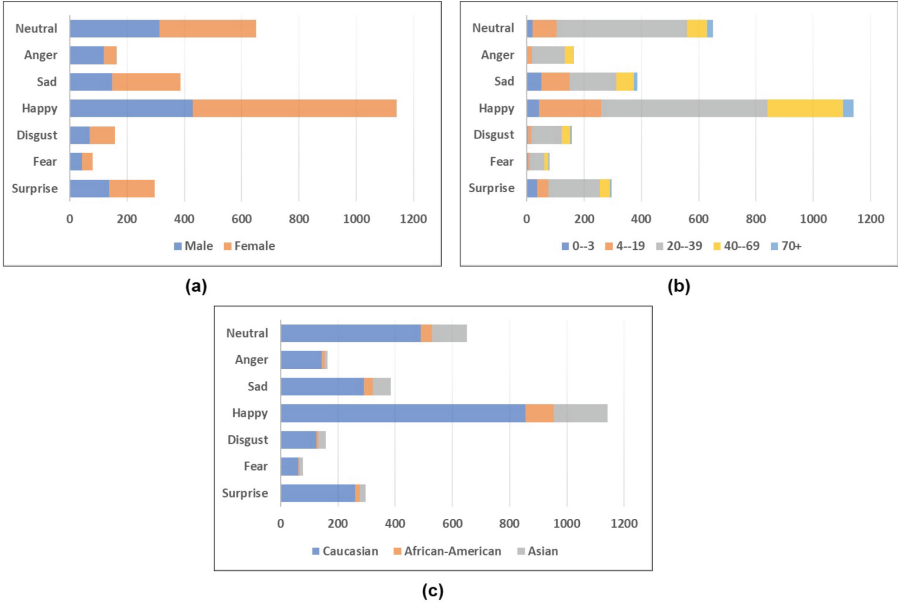
Our methodology and setup is based on the hypothesis that protected attributes can be forgotten without information loss of other facial attributes. Ideally, a network would be able to perfectly distinguish attributes if these attributes were completely separable from the rest of the informative features of the image. However, since they are not, we hypothesize that if subsets of a dataset partitioned on the basis of the protected attribute are aligned or brought closer in a latent space, these attributes are considered to be forgotten.

To achieve this, we use a discriminator module to classify the latents into their respective protected attributes. When this discriminator cannot determine membership of a latent into a particular protected attribute subset, then fairness can be achieved since the classification would be done solely on the basis of a latent which does not contain information about the protected attribute.

**Table 1.** Comparison of class wise accuracies on RAF-DB.

Emotion	Accuracy[%]	
	Ours	DA[42]
Happiness	92.0	93.3
Angry	83.2	81.0
Disgusted	57.7	54.1
Fearful	60.2	53.8
Surprised	82.9	81.8
Sadness	76.0	77.7
Neutral	81.0	82.1
<b>Mean</b>	<b>76.1</b>	<b>74.8</b>

**RAF-DB Bias Analysis.** Most FER datasets do not have the respective age, gender, and ethnic labels; therefore, to substantiate our results, we conducted experiments on RAF-DB [26], one of the most popular benchmark FER datasets.



**Fig. 3.** Data Distribution of the test set of RAF-DB. (a) represents the gender-wise distribution, (b) represents the age group distribution, and (c) represents the ethnic distribution of the test set of RAF-DB.

RAF-DB contains 15,339 images of diverse facial expressions downloaded from the internet and annotated manually by crowd-sourcing and reliable estimation; this dataset consists of seven basic expressions and eleven compound expressions.

RAF-DB provides labels that include expression, gender type, ethnicity, and age group. Fig. 3 showcases the attribute-wise breakdown of each label class in the test data. Since the distribution of test and training data is kept similar, we can draw few inferences from this distribution.

- Considering "race" as an attribute, we observe that almost 77% of the images belong to a single class i.e. Caucasian, rest, 23% are then distributed among two attributes, namely African-American and Asian.
- Similarly, for the age attribute, almost 57% of the images belong to one of the five age brackets, namely  $\{20-39\}$ . The rest of the 43% of images are distributed among the remaining four classes. Moreover, senior citizens from the 70+ age bracket and infants from  $\{0-3\}$  age bracket are highly under-represented, consisting of about 3% and 5% of the total images, respectively.
- Observing the expression attribute, we can infer that 39.7% of the total images belong to one of the seven expression classes, i.e. happy; the rest of the six classes are then distributed among the remaining six expressions. Moreover, expressions like fear, disgust and surprise are highly under-represented, consisting of about 2.7%, 5% and 10% of the total images, respectively.

**Table 2.** Mean expression-wise accuracy categorized by Gender and Race attributes on RAF-DB.

Attribute Labels	Mean Class wise Accuracies						
	DA[42]	Offline[10]	Focal Loss[27]	DDC[12]	DIC[41]	SS[13]	Ours
Male	74.2	72.0	71.0	71.0	72.0	72.0	<b>76.3</b>
Female	74.4	75.0	75.0	74.0	75.0	76.0	<b>76.0</b>
Caucasian	75.6	74.0	73.0	72.0	74.0	74.0	<b>76.15</b>
African-American	76.6	76.0	75.0	73.0	76.0	75.0	<b>77.1</b>
Asian	70.4	76.0	75.0	74.0	77.0	76.0	<b>75.5</b>

**Table 3.** Mean class-wise accuracy segmented by Age, Gender, and Race attributes on the RAF-DB dataset.

Attribute Labels	Mean Class wise Accuracies	
	DA[42]	Ours
0-3	80.2	<b>82.4</b>
4-19	69.9	<b>72.3</b>
20-39	76.4	<b>77.0</b>
40-69	74.4	<b>75.7</b>
70+	62.2	<b>70.1</b>
M-Caucasian	74.5	<b>76.0</b>
M-African-American	80.2	<b>81.1</b>
M-Asian	70.2	<b>73.4</b>
F-Caucasian	75.5	<b>76.2</b>
F-African-American	87.6	<b>81.1</b>
F-Asian	69.0	<b>71.7</b>

**Table 4.** Comparison of bias mitigation performance (where higher values indicate better outcomes) on RAF-DB, categorized by attribute labels.

Protected attributes	Mitigation of Bias						
	DA[42]	Offline[10]	Focal Loss[27]	DDC[12]	DIC[41]	SS[13]	Ours
Gender	<b>99.97</b>	95.4	96.1	96.2	95.4	95.4	99.51
Race	91.9	97.4	97.2	<b>97.6</b>	96.5	97.5	94.2
Age	82.1	-	-	-	-	-	<b>84.8</b>

This further substantiates our claim and establishes the need to mitigate bias in most FER datasets. The expression accuracy shown in Table 1 does not sufficiently portray the performance variation of classifiers across different demographics; therefore, in Table 2,3, we comprehensively compare accuracies broken down by each demographic group. Furthermore, to substantiate the interplay of "gender" and "race" attributes we also provide results of joint "Gender-Race" groups in Table 3. From Table 2,3 it can be inferred, that our proposed

method outperforms for mean class-wise accuracies broken down by attributes "age", "gender", "race" and "gender-race". To provide a numerical assessment of mitigation of bias for sensitive attributes such as age, gender, and race, in Table 4, we provide comparisons with [10, 12, 13, 27, 41, 42] using our evaluation metric for fairness (using Equation 4). From Table 4 we can infer that with regards to bias mitigation, our approach performs almost at par with Xu et al. [42] for "gender" attribute, whereas for "age" class it outperforms [42].

**Table 5.** Comparison of accuracy segmented by the smiling attribute in the CelebA dataset.

Expression	Accuracy	
	DA [42] [42]	Ours
Smiling	92.2	92.9
Not-Smiling	94.1	94.8
Mean	93.15	93.85

**Table 6.** Mean accuracy per class categorized by attributes on the CelebA dataset.

Attributes	Mean Class-wise Accuracy	
	DA[42][42]	Ours
Female	93.6	94.5
Male	91.9	93.4
Old	91.6	92.5
Young	93.6	94.3
Female-Old	92.7	93.3
Female-Young	93.8	94.9
Male-Old	90.7	92.1
Male-Young	92.8	93.7

### CelebA Bias Analysis

We conduct experimentation for images in CelebA for the binary attribute of "smiling". This is done to facilitate the expression recognition of happy. We use the CelebA dataset since it is much larger as compared to RAF-DB with 39920 images in a subset of CelebA as compared to 12271 in all of RAF-DB. The protected attributes we use for fairness are Gender and Age.

The Smiling/No Smiling attribute is evenly distributed with exactly 50% of the images having the smiling attribute. The image distribution for Gender and Age are not evenly distributed, with a 22.8% gap between the number of Male and Female images, and a 51.4% gap between the number of Young and Old images. The comparison of accuracies with "Smiling" vs "No Smiling" is



shown in Table 5. Since this is a simple binary classification task, accuracies are almost comparable. Table 6 provides comparable class-wise (i.e. "Smiling" vs "No Smiling") accuracies broken down by attribute labels ("gender", "age", and "Gender-Age"). Table 7 provides comparisons with [42] using our evaluation metric for fairness (using Equation 4) on sensitive attributes.

**Table 7.** Comparison of bias mitigation (where higher values indicate better performance) on CelebA, categorized by attribute labels.

Protected Attribute	Bias Mitigated	
	DA[42][42]	Ours
Gender	98.3	99.1
Age	98.1	98.9
Gender-Age	96.9	98.0

## 5 Ablation Study

We demonstrate the importance and effectiveness of each technical contribution through this ablation study on RAF-DB [26]. We first look at the impact of using a Variational Autoencoder as compared to a standard Autoencoder or other dimensional reduction techniques. We can see a significant drop in accuracy and a corresponding drop in bias mitigation when an Autoencoder is used in place of a VAE. We believe this is due to the ability of VAEs to generate denser representations due to the KL-Divergence loss from the Gaussian distribution present in VAEs.

**Table 8.** Component-wise Ablation Study of our model.

Component	Mean Accuracy	Bias (Gender)	Bias (Race)
<b>VAE+MBCConv+Discriminator (Ours)</b>	76.1	99.93	94.2
Auto Encoder+Discriminator+MBCConv	74.2	97.6	91.2
VAE+Discriminator+ResBlock	74.5	99.91	93.8
VAE+MBCConv	76	91.4	79.2
VAE+ResBlock	73	91	79.3

We further look at the impact of the Discriminator module on latent space alignment and examine whether it increases fairness. From Table 8, we see that there is a significant decrease in bias mitigation when the VAE is trained without the min-max objective jointly with the discriminator. This demonstrates that the Discriminator is highly impactful for latent space alignment and that

the sensitive attributes are encoded in the latent without it.

We further analyze the impact of the CNN classifier backbone on accuracies. We find that the MBConv block[18] performs superior as compared to ResBlock [16]. In recent works, MBConv blocks have been known for their superior expressive power in CNNs. MBConv outperforms ResBlocks given all other parameters remain the same. However, this difference is minimal given that the largest contributor to our model is the VAE+Discriminator architecture for latent alignment.

## 6 Conclusion

With the exponential increase of real-world artificial intelligence systems deployed in our daily lives, accounting for fairness has become a crucial factor in the design and research of such systems. AI systems can be deployed in various critical settings to make important life-changing decisions; hence, ensuring that these decisions do not exhibit bias or discriminatory behaviour against specific groups or demographics is of utmost importance. As a result, bias mitigation investigation and its mitigating strategies have gained good traction among researchers. Recently, many works have proposed bias mitigation strategies through traditional machine learning and deep learning in various subdomains; however, this is a relatively less-explored area in facial expression recognition. In this research, we present an innovative approach to reducing bias in FER systems by integrating a Variational Autoencoder with an Adversarial Discriminator, followed by a classification module utilizing MBConv. Our method not only surpasses the results reported in [42] but also introduces a versatile framework that can be adapted for other image classification tasks. To our knowledge, this is the first work to leverage latent alignment for bias mitigation in FER systems. We aim for our research to pave the way for more extensive exploration of latent space manipulation in addressing bias across diverse image classification challenges.

## References

1. Alvi, M., Zisserman, A., Nellåker, C.: Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)
2. Beigman, E., Klebanov, B.B.: Learning with annotation noise. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 280–287 (2009)
3. Buolamwini, J., Gebru, T.: Gender shades: Intersectional accuracy disparities in commercial gender classification. In: Conference on fairness, accountability and transparency. pp. 77–91. PMLR (2018)
4. Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., Varshney, K.R.: Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems* **30** (2017)

5. Celis, L.E., Huang, L., Keswani, V., Vishnoi, N.K.: Classification with fairness constraints: A meta-algorithm with provable guarantees. In: Proceedings of the conference on fairness, accountability, and transparency. pp. 319–328 (2019)
6. Chen, Y., Joo, J.: Understanding and mitigating annotation bias in facial expression recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14980–14991 (2021)
7. Chen, Y., Joo, J.: Understanding and mitigating annotation bias in facial expression recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14980–14991 (2021)
8. Chen, Y., Joo, J.: Understanding and mitigating annotation bias in facial expression recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14980–14991 (2021)
9. Chen, Z., Zhang, J.M., Sarro, F., Harman, M.: A comprehensive empirical study of bias mitigation methods for machine learning classifiers. *ACM transactions on software engineering and methodology* **32**(4), 1–30 (2023)
10. Churamani, N., Kara, O., Gunes, H.: Domain-incremental continual learning for mitigating bias in facial expression and action unit recognition. *IEEE Trans. Affect. Comput.* **14**(4), 3191–3206 (2022)
11. Drozdowski, P., Rathgeb, C., Dantcheva, A., Damer, N., Busch, C.: Demographic bias in biometrics: A survey on an emerging challenge. *IEEE Transactions on Technology and Society* **1**(2), 89–103 (2020)
12. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference. pp. 214–226 (2012)
13. Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
14. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 259–268 (2015)
15. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. *Advances in neural information processing systems* **29** (2016)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hort, M., Chen, Z., Zhang, J.M., Harman, M., Sarro, F.: Bias mitigation for machine learning classifiers: A comprehensive survey. *ACM Journal on Responsible Computing* (2023)
18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
19. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
20. Jia, S., Lansdall-Welfare, T., Cristianini, N.: Right for the right reason: Training agnostic networks. In: Advances in Intelligent Data Analysis XVII: 17th International Symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, October 24–26, 2018, Proceedings 17. pp. 164–174. Springer (2018)

21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. pp. 694–711. Springer (2016)
22. Kamiran, F., Karim, A., Zhang, X.: Decision theory for discrimination-aware classification. In: *2012 IEEE 12th international conference on data mining*. pp. 924–929. IEEE (2012)
23. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Fairness-aware classifier with prejudice remover regularizer. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24–28, 2012. Proceedings, Part II 23*. pp. 35–50. Springer (2012)
24. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
25. Li, S., Deng, W.: Deep facial expression recognition: A survey. *IEEE Trans. Affect. Comput.* **13**(3), 1195–1215 (2020)
26. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2584–2593. IEEE (2017)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2980–2988 (2017)
28. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: *Proceedings of the IEEE international conference on computer vision*. pp. 3730–3738 (2015)
29. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)* **54**(6), 1–35 (2021)
30. Morales, A., Fierrez, J., Vera-Rodriguez, R., Tolosana, R.: Sensitivenets: Learning agnostic representations with application to face images. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(6), 2158–2164 (2020)
31. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2009)
32. Parraga, O., More, M.D., Oliveira, C.M., Gavenski, N.S., Kupssinskü, L.S., Medronha, A., Moura, L.V., Simões, G.S., Barros, R.C.: Fairness in deep learning: A survey on vision and language research. *ACM Computing Surveys* (2023)
33. Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., Weinberger, K.Q.: On fairness and calibration. *Advances in neural information processing systems* **30** (2017)
34. Rhue, L.: Racial influence on automated perceptions of emotions. Available at SSRN 3281765 (2018)
35. Rizvi, S.S.A., Seth, A., Narang, P.: Fair-fer: A latent alignment approach for mitigating bias in facial expression recognition (student abstract). In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 23633–23634 (2024)
36. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 4068–4076 (2015)
37. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 7167–7176 (2017)
38. Wang, M., Deng, W., Hu, J., Tao, X., Huang, Y.: Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 692–702 (2019)

39. Wang, T., Zhao, J., Yatskar, M., Chang, K.W., Ordonez, V.: Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5310–5319 (2019)
40. Wang, Z., Qinami, K., Karakozis, I.C., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8919–8928 (2020)
41. Wang, Z., Qinami, K., Karakozis, I.C., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8919–8928 (2020)
42. Xu, T., White, J., Kalkan, S., Gunes, H.: Investigating bias and fairness in facial expression recognition. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 506–523. Springer (2020)
43. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: International conference on machine learning. pp. 325–333. PMLR (2013)
44. Zeng, D., Lin, Z., Yan, X., Liu, Y., Wang, F., Tang, B.: Face2exp: Combating data biases for facial expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20291–20300 (2022)
45. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340 (2018)
46. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society. pp. 335–340 (2018)
47. Zhuang, H., Young, J.: Leveraging in-batch annotation bias for crowdsourced active learning. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining. pp. 243–252 (2015)



# DDCTrack: Dynamic Token Sampling for Efficient UAV Transformer Tracking

Guocai Du<sup>1</sup>, Peiyong Zhou<sup>1</sup>, Nurbiya Yadikar<sup>1</sup>, Alimjan Aysa<sup>1,2</sup>,  
and Kurban Ubul<sup>1,2</sup>✉

<sup>1</sup> College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China  
kurbanu@xju.edu.cn

<sup>2</sup> The Key Laboratory of Multilingual Information Technology, Xinjiang University,  
Urumqi 830046, China

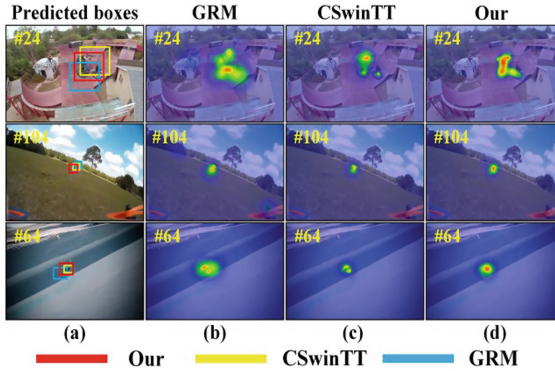
**Abstract.** Although state-of-the-art transformer models have shown promising results in unmanned aerial vehicle (UAV) tracking, they come with high computational demands. Existing tracking methods aim to reduce computational complexity by controlling the number of tokens. However, this method is not effective for all tracking methods. Therefore, we propose a novel dynamic token sampling for an efficient UAV transformer tracking framework. Unlike previous transformer-based tracking methods, our method avoids the need for complex head networks like classification and regression. It solely employs our newly designed encoder, comprising three key components: Dynamic Position Embedding, Dynamic Token Sampler, and Convolutional Feed-Forward Network. This module enhances visual representation by scoring and dynamically sampling tokens, allowing for a flexible token count that adapts to target changes within each frame. We utilize a simple image-sequence contrastive loss as the loss function. Our approach not only simplifies the tracking framework, but also achieves state-of-the-art performance on multiple challenging datasets at real-time run speeds.

**Keywords:** Unmanned Aerial Vehicle · Dynamic Token Sampling · Image-sequence Contrastive Loss · Real-time Tracking

## 1 Introduction

Visual object tracking is one of the fundamental tasks in computer vision. It involves tracking the position of a chosen object solely based on an initial frame, across subsequent frames. Due to its extensive application in the field of unmanned aerial vehicles (UAV) [1], such as aerial cinematography, collision warning, and visual localization, it has attracted widespread attention. Recently, prevalent tracking methods mostly employ a divide-and-conquer strategy, breaking down the tracking problem into multiple sub-problems, such as feature extraction and relation modeling. These sub-problems are handled by specific networks. The mainstream methods primarily address the issue through a two-stage and two-stream pipeline [2]. Here, two-stage refers to decomposing the tracking process into two stages: feature extraction and relationship modeling.

Two-stream involves processing search images and template images separately. These divide-and-conquer strategies have proven to achieve significant performance in tracking benchmarks, consequently becoming the design of the current mainstream models.



**Fig. 1.** Visualization of Early Attention Maps for Different Methods.

However, recent research has identified shortcomings in feature extraction and relation modeling. Firstly, based on the Transformer method, extracting shallow features results in high redundancy. As depicted in Fig. 1(c), shallow attention focuses more on adjacent tokens given an anchor token, paying less attention to distant tokens. Consequently, global comparisons between tokens in subsequent processes lead to increased computational complexity in capturing local correlations. To address this issue, GRM [3] proposed an adaptive token that offers more flexible modeling capabilities, reducing attention on local regions. As shown in Fig. 1(b), GRM moderately reduces redundancy in local attention, focusing on only a few surrounding tokens. Then, CSwinTT [4] performs feature extraction without prior knowledge of the object. Specifically, image feature extraction is determined post off-line training, resulting in weak interactions between the template and search region. Finally, despite the outstanding expressive capabilities of the transformer, it suffers from the drawback of high computational costs. The computational cost is quadratically related to the number of tokens used. Hence, an essential need exists to effectively reduce the number of tokens to lower computational expenses. OTrack [31] proposed a candidate elimination module that retains the top-k corresponding candidate weights, reducing computational costs. However, a fixed approach undoubtedly leads to the loss of useful information.

To address these issues, we propose a new dynamic token sampling for efficient UAV transformer tracking framework (DDCTrack), as illustrated in Fig. 3. The motivation behind our approach lies in the observation that the contribution of information from the object and search region to the final tracking varies significantly, containing a considerable amount of redundant and irrelevant data. The tracked object determines the quantity of relevant information. As shown in Fig. 5, it is clear that only a specific number of markers are required to achieve accurate target tracking, and that this number varies at each stage. Therefore, we introduce a method capable of dynamically selecting the minimum required tokens according to the object at different stages. This approach

is entirely different from EViT [5], which specifies the selection of tokens based on a fixed ratio during training. Such a static approach risks losing critical information, particularly in challenging tracking datasets. It can also lead to unnecessary token wastage in simpler tracking scenarios, increasing computational costs. We reduce unnecessary waste by dynamically adjusting the number of tokens. Additionally, to enhance interaction between information, the flattened template and search region can be directly concatenated, boosting target discrimination. This direct connection between the template and search region facilitates highly parallelized tracking, eliminating the need for additional networks for feature extraction, striking a favorable balance between speed and performance.

Moreover, our model not only takes images as inputs but also transforms four supervised values into discrete tokens. By introducing the proposed DDC encoder, which combines vision and coordinates, it enhances the visual representation. Training is conducted on contrastive loss using image-sequence pairs, eliminating the need for further fine-tuning. Despite our framework’s simplicity, the proposed tracking performance demonstrates impressive results, reaching new SOTA levels across multiple datasets. Compared to other transformer-based trackers, we maintain superior inference efficiency and faster convergence. It is worth noting that existing methods heavily rely on intricately designed head networks or complex loss functions. However, our tracker utilizes only two encoder structures with a simple loss function. DDCTrack achieves a favorable balance between speed and accuracy, as depicted in Fig. 8.

In summary, our work primarily involves the following contributions:

- (1) We have designed a novel UAV tracking framework that introduces a new perspective to tracking by utilizing sequences as supervision.
- (2) We proposed a new DDC module comprising three crucial components: Dynamic Position Embedding (DPE), Dynamic Token Sampler (DTS), and Convolutional Feed-Forward Network (ConvFFN). Specifically designed for shallow features, it effectively learns global representations. Moreover, it dynamically adjusts the number of tokens and is a differentiable parameter-free module.
- (3) Through comprehensive experiments across multiple datasets, we validated that our proposed approach exhibits superiority in terms of inference speed, tracking performance, and convergence speed when compared to existing methods.

## 2 Related Work

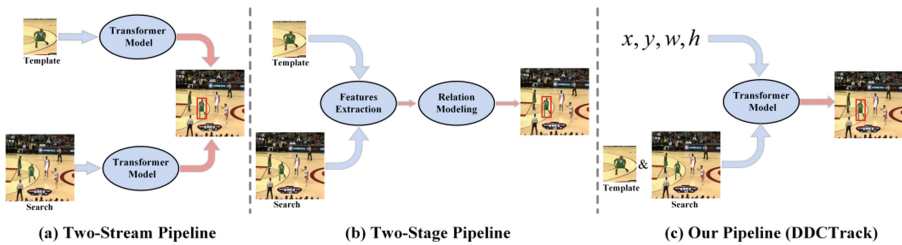
### 2.1 Vision Transformers

As transformer models achieved remarkable success in natural language processing [6], ViT introduced transformer architecture into computer vision, resulting in groundbreaking achievements. Leveraging its advantage in modeling long-range dependencies, many researchers have started focusing on designing visual transformers. Various approaches, such as self-attention variants, novel hierarchical architectures, and positional encodings, have been applied to visual tasks [7]. However, computations based on transformer architectures are often determined by the number of tokens, inevitably leading to increased computational costs. Consequently, several effective self-attention mechanisms have



been introduced to alleviate these computational burdens. For instance, PVT [8] introduced a pyramid architecture with downsampling key and value tokens. Reformer [9] employed hashing functions to allocate tokens into buckets and applied dense attention within each bucket. Orthogonal Transformer [10] computed an orthogonal space to represent global and local features. ACT [11] and TCFormer [12] treated merged tokens as queries and the original tokens as keys and values, aiming to reduce computational costs. In contrast to these methods, our proposed DDC utilizes soft associations to establish sparse mappings between tokens and super tokens, employing self-attention in the super token space.

## 2.2 Visual Tracking



**Fig. 2.** Three different tracking pipeline.

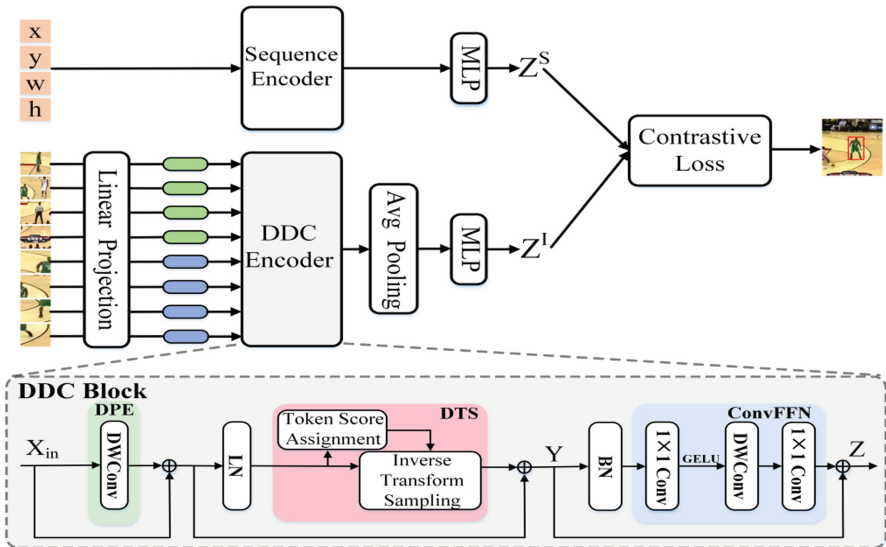
As depicted in Fig. 2, the two-stream pipeline [13] involves using a transformer model to extract features from both the template and the search region. The two-stage pipeline sequentially divides the process into two steps: feature extraction and relation modeling. Based on the above differences, our method is compared with the above different structural prototypes. Earlier methods and some transformer-based tracking methods belong to the category shown in Fig. 2(a). These techniques extract features from the template and search regions separately using a backbone. However, they lack the ability to adjust template features based on the search region and struggle with effective relationship modeling. Therefore, the two-stage approach was introduced, incorporating feature extraction and relation modeling (for example, siamese utilizing cross-correlation operations and transformer self-attention mechanism). This led to a relatively complex relationship module, as shown in Fig. 2(b). STARK [14] concatenated the search region with pre-extracted template features and fed them into multiple self-attention layers. TransT [15] stacked a series of self-attention and cross-attention layers for feature fusion. While the two-stream structure improves performance, it inevitably sacrifices speed. In contrast, our structure is different, as depicted in Fig. 2(c). Firstly, it combines template features and the search region as input into a transformer model and then integrates the object’s coordinate position into a single framework. Our pipeline efficiently provides features and relationship modeling at a lower cost, guiding each other to generate the final object position efficiently in both training and testing phases.

### 2.3 Sequence Learning with Text Supervision

Through large-scale image-text pairs datasets, representation learning with text-supervised methods [16] has been applied to various visual tasks, such as object detection and segmentation. Moreover, in cross-modality domains, sequence learning has been integrated. For instance, Flamingo and DALL-E [17] have adopted sequence-to-sequence learning to unify multi-modality tasks.

Referring to PixSeq, our sequence learning shares similarities. Both methods leverage sequence generation to address visual tasks and discretize continuous values of bounding box coordinates into integers. However, there are differences compared to Pix2Seq. Firstly, the representation of sequences differs. Pix2Seq expresses sequences through object corner coordinates and categories, whereas we utilize center point coordinates, width, and height for representation. Secondly, the methodologies differ. Pix2Seq utilizes ResNet and encoder-decoder transformer. In contrast, our method is simpler, relying solely on our proposed DDC encoder. Thirdly, the task objectives vary. Pix2Seq is designed for object detection, while our objective is tracking. Furthermore, our method is end-to-end, allowing seamless integration of earlier tracking designs like online template updates into our tracking approach.

## 3 Method



**Fig. 3.** Architecture of the proposed DDCTrack. The key component is an DDC encoder, which consists of DPE, DTS, and ConvFFN, respectively.

In this section, we propose the DDCTrack architecture for UAV tracking with sequence supervision, as depicted in Fig. 3. Initially, we introduce the sequence encoder. Subsequently, we provide a detailed description of the proposed DDC encoder module.

Finally, we introduce the image-sequence contrastive loss and integrating online update techniques.

### 3.1 Sequence Encoder

We first convert the object bounding box into a sequence of discrete tokens. Typically, a bounding box is determined by its center position  $(x, y)$ , width, and height  $(w, h)$ . There are various representations for expressing the bounding box, such as  $[w, h, x, y, ]$  and  $[x, y, w, h]$ . From an intuitive standpoint,  $[x, y, w, h]$  aligns more with human prior knowledge, implying the prioritization of locating the object position before estimating its width and height. We normalize these continuous coordinates into integers between  $[1, n_{bins}]$ . The integers between  $[1, n_{bins}]$  are considered as a word within  $V$ . Experimental findings indicate higher precision when  $n_{bins}$  is set to 3500 (detailed in Sect. 4.3). Each word in  $V$  has a corresponding learnable embedding, which is optimized during training. In the final stage of the DDCTrack model, we compute the contrastive loss for image-sequence pairs using formula  $\mathcal{S}_{\text{ground}}$ .

### 3.2 DDC Encoder

The image encoder is a transformer-based architecture designed by us to represent visual features. Firstly, a linear projection converts search image patches and template image patches into visual embeddings, and these visual embeddings are then fed into transformer layers and concatenated together. Subsequently, they pass through DDC blocks for representation extraction. Finally, average pooling is applied to the output to obtain the global representation of the object.

The DDC Block consists of three components: DPE, DTS, and ConvFFN.

$$X = DPE(X_{in}) + X_{in} \quad (1)$$

$$Y = DTS(LN(X)) + X \quad (2)$$

$$Z = \text{ConvFFN}(\text{BN}(Y)) + Y \quad (3)$$

**Dynamic Position Embedding.** We dynamically incorporate position information into all tokens using DPE (Eq. 1) to effectively leverage the spatiotemporal order of tokens for object modeling. In contrast to convolutional position embedding, relative positional encoding, and absolute position encoding [18], DPE overcomes permutation invariance and is resolution-agnostic. This is due to its shared convolutional parameters, locality, and zero padding, aiding tokens along the object boundaries to discover their absolute position. Consequently, all tokens can encode their absolute spatiotemporal position merely by querying their neighbors. Our DTS efficiently explores and utilizes long-range dependency relationships to extract contextual representations. A detailed description of DTS will be provided in the subsection below. Furthermore, ConvFFN enhances local feature representation, comprising a  $3 \times 3$  depth-wise convolution, a non-linear function (such as GELU), and two  $1 \times 1$  convolutions. It is noteworthy that both ConvFFN and DPE

utilize depthwise convolutions to reinforce the learning capacity of local features. Meanwhile, DTS effectively employs long-range dependencies to extract global contextual features. Therefore, the combined utilization of these three components significantly improves our model’s ability to capture both local and global dependencies.

**Dynamic Token Sampler.** Due to the computation in transformers being determined by the number of tokens, many SOTA vision transformers are computationally expensive, and the number of tokens remains static at each stage. Convolutional neural networks typically reduce parameter counts through various pooling operations to mitigate computational expenses. This often leads to a direct reduction in spatial or temporal resolution within the network. However, applying such fixed-kernel pooling operations directly to vision transformers is not straightforward. The reason being, tokens are permutation-invariant, and employing fixed downsampling operations is not an optimal choice. On one hand, fixed downsampling may cause the loss of crucial information for the target in certain video frames. On the other hand, it results in many irrelevant features for object tracking. Therefore, we propose a DTS capable of dynamically adjusting the number of tokens in each stage of the transformer. This overcomes the issue of losing critical information for the target while reducing computational resources.

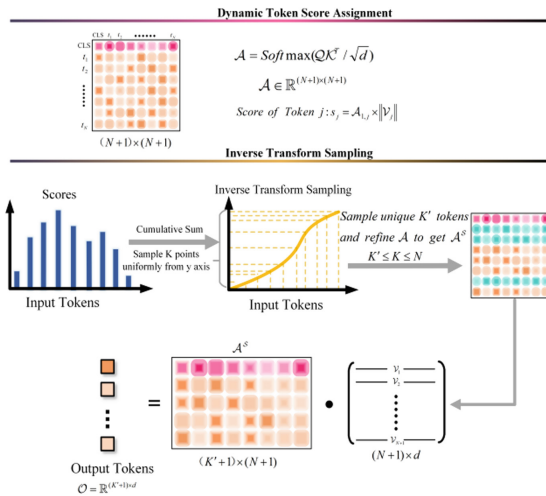
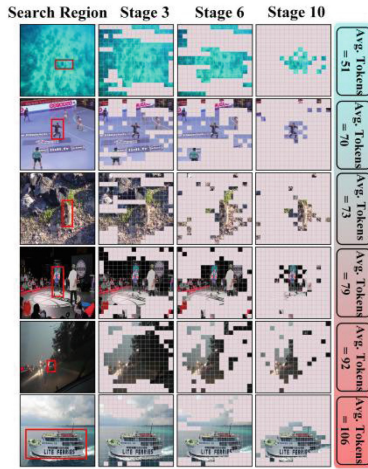


Fig. 4. Flow of the DTS Module.

DTS is a parameter-free differentiable module that samples tokens containing crucial information based on the input tokens, as depicted in Fig. 4. In the DTS module, token scores are calculated using token scoring for each token, and then a subset of these tokens is sampled based on these scores.

**Token Scoring.** Given the input token  $X \in \mathbb{R}^{(N+1) \times d}$ , there is a self-attention layer with  $N + 1$  tokens. The ViT first connects the classification token to the input token, and then processes it through the model. Finally, the output tokens corresponding to the last transformer block are fed to the classification head to obtain the classification probabilities. Our goal is to reduce the output tokens  $\mathcal{O} \in \mathbb{R}^{(K'+1) \times d}$ , dynamically

adjusting based on the input image (where  $K'$  represents the number of sampled tokens), while meeting  $K' \leq K \leq N$ , where  $K$  is the parameter for the maximum sampled quantity. The quantity of sampled tokens  $K'$  varies based on different stages of the network and data variations, as illustrated in Fig. 5. The scoring criterion for each token is as follows.



**Fig. 5.** Visualization of Dynamic Token Sampling Process.

In the standard self-attention layer, the queries  $Q \in \mathbb{R}^{(N+1) \times d}$  and key  $\mathcal{K} \in \mathbb{R}^{(N+1) \times d}$  and values  $\mathcal{V} \in \mathbb{R}^{(N+1) \times d}$  are calculated by input tokens  $X \in \mathbb{R}^{(N+1) \times d}$  respectively. The queries and keys undergo a dot product operation to obtain the attention matrix  $\mathcal{A}$ , which is scaled down by a factor of  $\sqrt{d}$ .

$$\mathcal{A} = \text{Soft max}(Q\mathcal{K}^T / \sqrt{d}) \quad (4)$$

Then, the output tokens are computed by a combination of the values weighted by the attention weights.

$$\mathcal{O} = \mathcal{A}\mathcal{V} \quad (5)$$

where each row of  $\mathcal{A}$  contains the attention weights for each output token, indicating the contribution of input tokens to the output tokens. The  $\mathcal{A}_{1, :}$  contains the classification token, where  $\mathcal{A}_{1, j}$  represents the input tokens and  $j$  denotes the importance for the output classification token. Therefore, we filter the attention matrix  $\mathcal{A}$  by using  $\mathcal{A}_{1, 2}, \dots, \mathcal{A}_{1, N+1}$  as significance scores, as specifically described in Fig. 4. Here, to preserve the classification token, we did not utilize  $\mathcal{A}_{1, 1}$ . In Eq. 5, it can be observed that the output tokens  $\mathcal{O}$  are determined by both  $\mathcal{A}$  and  $\mathcal{V}$ , thus introducing the norm of  $\mathcal{V}_j$  to calculate the significance score for the  $j$ -th token. The reason is that the smaller the norm, the lesser the impact, indicating the corresponding token is less significant. The ablation experiments demonstrated that the norms of the  $\mathcal{A}_{1, j}$  and  $\mathcal{V}_j$  contribute to improving the tracking

results. The calculation method for the significance score of token  $j$  is as follows.

$$S_j = \frac{S_{1,j} \times \|\mathcal{V}_j\|}{\sum_{i=2} \mathcal{A}_{1,i} \times \|\mathcal{V}_i\|} \quad (6)$$

where  $i, j \in \{2 \dots N\}$ . For multi-head attention layers, the scores for each head are computed, followed by an addition of these head scores.

**Token Sampling.** According to Eq. 6, to compute the significance scores of all tokens, we can select the corresponding rows through the attention matrix  $\mathcal{A}$ . A straightforward method is to choose the top  $K$  tokens with higher scores. However, from the experiments (detailed in Sect. 4.3), it can be concluded that this method is not optimal and does not dynamically select the top  $K$  tokens. We analyze the reason, which could be due to directly discarding tokens with lower scores, resulting in the potential loss of useful information. Particularly, in cases where the discriminative capability is weak, some information might not be extracted. For instance, in the early stages, the softmax function might cause a reduction in attention weights for multiple tokens with similar keys. Therefore, it is possible to sample tokens based on their significance scores, where the probability of sampling one among similar tokens is proportional to their summed scores.

During the sampling stage, inverse transform sampling is used based on the tokens' significance scores. These scores are normalized, hence they can be interpreted as probabilities. The cumulative distribution function (CDF) is computed as follows, starting from the second token as with token scoring. Once the cumulative distribution function is available, the inverse operation of the CDF is applied to obtain the sampling function.

$$\Psi(k) = CDF^{-1}(k) \quad (7)$$

$$CDF_i = \sum_{j=2}^{j=i} S_j \quad (8)$$

where  $k \in [0, 1]$ . It can be concluded that significance scores can calculate the map function between original tokens and sampled tokens. We can sample  $K$  times from the uniform distribution  $U [0, 1][0, 1]$  to obtain  $K$  samples. Such randomization may be desirable in some areas, but deterministic inference takes precedence in most cases. As a consequence, a fixed sampling approach of  $k = \{\frac{1}{2K}, \frac{3}{2K}, \dots, \frac{2K-1}{2K}\}$  is chosen for both training and inference. Due to  $\Psi(\cdot) \in$ , the indices of the nearest significant scores' tokens are taken as sampling indices.

When a token is sampled multiple times, it only needs to be retained once. Therefore, the quantity of unique indices  $K'$  is far less than  $K$ . From Fig. 5, it can be observed that in the earlier stages, more tokens are selected, indicating lower feature discrimination ability and more balanced attention weights. However, in later stages, the situation is reversed. The number and position of tokens also vary depending on different images. When the background containing the object is relatively clean, only a few tokens are sampled. Conversely, in cluttered backgrounds, more tokens are required. This validates the significance of our proposed dynamic token sampling method.

The indices of sampled tokens can be used to correct the attention matrix  $\mathcal{A} \in \mathbb{R}^{(N+1) \times (N+1)}$  by selecting the rows corresponding to the sampled  $K' + 1$  tokens.  $\mathcal{A}^S \in$

$\mathbb{R}^{(K'+1) \times (N+1)}$  denotes the corrected attention matrix. Replace  $\mathcal{A}$  in Eq. 5 with  $\mathcal{A}^s$  to obtain output tokens  $\mathcal{O} \in \mathbb{R}^{(K'+1) \times d}$ .

$$\mathcal{O} = \mathcal{A}^s \mathcal{V} \quad (9)$$

### 3.3 Learning from Image-Sequence Pairs

To effectively train the DDCTrack model, we employ an image-sequence pairs contrastive loss between image-sequence pairs, described as follows.

**Image-Sequence Contrastive Loss.** To better learn visual representation through sequence supervision, we train the image-sequence contrastive loss using a dual-encoder model. Initially, the DDC encoder acts as the image encoder, generating visual features, while BERT functions as the sequence encoder, generating sequences. Both the image and sequence pairs are input into their respective encoders, projected into a common embedding space, and their similarity measures are computed. Subsequently, the successfully matched image-sequence pairs are considered as positive pairs, while the unsuccessful pairs are regarded as negative pairs. Finally, we pull positive pairs closer together and push unmatched negative pairs farther away.

In our approach, we calculate alignment scores  $S_{Ground}$  between the image and sequence.

$$O = Enc_I(Img), P = Enc_L(Coordinate), S_{ground} = OP^T \quad (10)$$

where  $P \in \mathbb{R}^{M \times d}$  represents the sequence embedding from the text encoder, acting similarly to the weight matrix in self-attention mechanisms.

## 4 Experiments

### 4.1 Implementation Details

**Model.** We used DeiT-S [19] + DDC as the visual encoder for DDCTrack-B256 and B384, and DeiT-B + DDC as the visual encoder architecture for DDCTrack-S256 and B384. The sequence encoder employed BERT [20]. Pre-training utilized ImageNet for initializing visual encoder parameters, with patch sizes set at  $16 \times 16$ . It is worth noting that cropping operations were not used to prevent disrupting the alignment of image and sequence signals. The vocabulary size and the quantization  $n_{bins}$  quantity are both set to 3500. The encoder has 8 attention heads, a hidden size of 256, and the Feed Forward Network has a hidden size of 1024. The word embedding dimension is the same as the decoder’s hidden size.

**Training.** Our training data consist of Youtube-BB, GOT-10K, COCO, and ImageNet VID. We trained the model using the AdamW [21] optimizer and set the learning rate and weight decay for both visual and sequence encoders to  $10^{-4}$ . The model was trained for 500 epochs with a warm-up strategy, with each epoch containing 60k image pairs. After 400 epochs, the learning rate was reduced by a factor of 10.

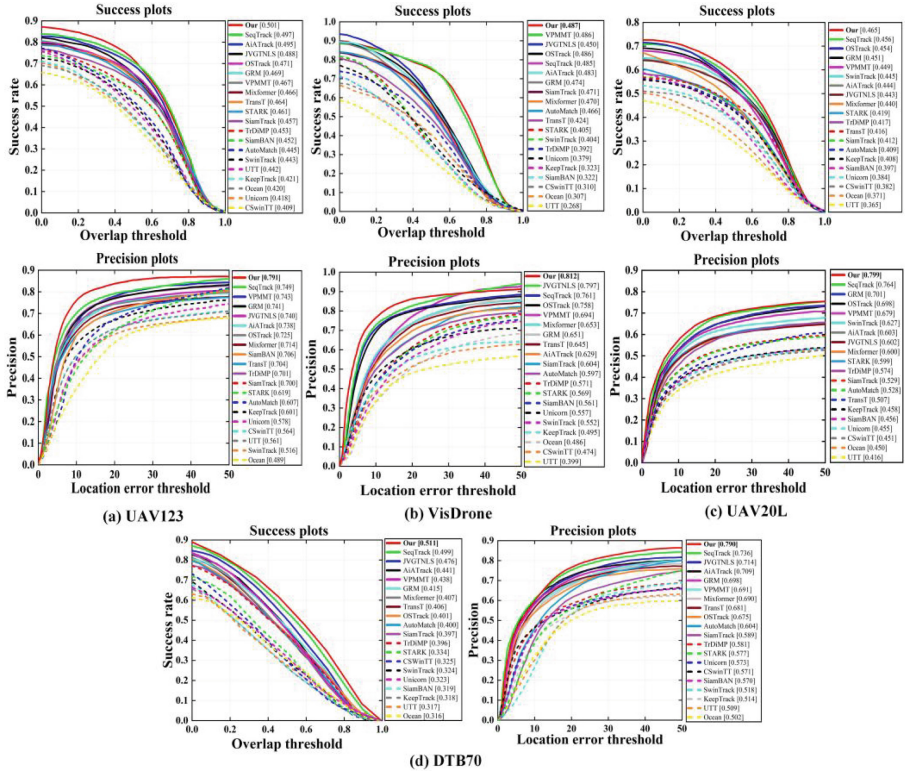


Fig. 6. Overall performance of all trackers on four well-known aerial tracking benchmarks.

## 4.2 Comparison with State-of-the-Art Trackers

For a comprehensive analysis, a comparative evaluation of DDCTrack and a number of SOTA trackers [22–34] was conducted on our authoritatively challenging public UAV dataset.

**UAV123.** The dataset consists of a large-scale UAV tracking benchmark comprising 123 high-quality challenging sequences, totaling over 112,000 frames. As shown in Fig. 6(a), our tracker outperforms other algorithms in both precision and success metrics. On precision, our tracker ranks first, surpassing the second-ranked SeqTrack (by 4.2 points) and the third-ranked VPMPT (by 4.8 points). On the success metric, DDCTrack exhibits improvement over the second-ranked SeqTrack (by 0.4 points), claiming the top position.

**VisDrone.** For VisDrone, it is an extensive dataset comprising over 20,000 images and more than 6 million annotated bounding boxes. As depicted in Fig. 6(b), our tracker achieves a precision higher by 1.5 points compared to the second-ranked JVGTLNS. Additionally, the success score slightly surpasses the VPMPT method by 0.1 points.

**UAV20L.** UAV20L consists of 20 long-term tracking sequences, totaling over 58,000 frames, with an average of approximately 2,934 frames per sequence. As illustrated in Fig. 6(c), our DDCTrack demonstrates superior performance compared to other SOTA



methods, underscoring the effectiveness of our proposed tracking framework. For example, in terms of accuracy, our method outperforms the second ranked SeqTrack and third ranked GRM by 3.5 and 9.8 points, respectively. Similarly, in terms of success rate, DDCTrack achieves the best results, followed by SeqTrack and OTrack, surpassing them by 0.6 and 1.1 points, respectively. These excellent experimental results validate that our tracker can be a top choice for long-term aerial tracking scenarios.

**DTB70.** Compared to the previous two datasets, the DTB70 dataset comprises a considerable number of scenes with fast motion, encompassing 70 challenging UAV sequences. As depicted in Fig. 6(d), DDCTrack achieves the best performance in both precision and success metrics. SeqTrack follows as the second-best performer, followed by JVGTNLS. Our method not only enhances precision but also improves speed. The primary reason is that our proposed approach efficiently samples critical tokens dynamically.

### 4.3 Ablation Study

In this section, we demonstrate the effectiveness of the proposed method from various perspectives. The experiments follow the one-pass evaluation (Precision and Success), using DDCTrack-S256 as the baseline model.

**Table 1.** Ablation Study on UAV123 and DTB70.

#	Method	UAV123	DTB70
1	Baseline	68.6	66.4
2	Joint $\rightarrow$ Separate	61.1	59.8
3	$[x, y, w, h] \rightarrow [w, h, x, y]$	67.3	65.7
4	$[x, y, w, h] \rightarrow [x_{lt}, x_{rb}, y_{lt}, y_{rb}]$	67.5	65.9
5	Concat of Search-Template	68.7	66.5
6	Avg. Of Search-Template	68.5	66.4
7	+ Integrating Online Update	71.8	70.9
8	+ Temporal	72.1	73.6

**Joint v.s. Separate.** The input to the image DDC encoder commonly employs two different approaches. One involves feeding both the template and search regions into the encoder jointly, extracting features together in a unified manner. Another approach is to refer to the Siamese method and extract the features of the template and search area separately, as shown in Table 1 (2). Experimental results on both datasets indicate a performance drop when features are extracted separately compared to the joint feature extraction method. We hypothesize that the joint method enables the encoder to effectively learn the complex correspondences between template and search images.

**The Encoder’s Inputs Differ.** We conducted a comparative analysis of different inputs for the sequence encoder, as shown in Table 1 (3 and 4). One approach is  $[w, h, x, y]$ , where the decoder initially generates the width and height  $[w, h]$  of the target and

subsequently produces the center position  $[x, y]$  of the target. Another approach is  $[x_{lt}, x_{rb}, y_{lt}, y_{rb}]$ , representing the top-left coordinate  $[x_{lt}, y_{lt}]$  of the target and the bottom-right coordinate  $[x_{rb}, y_{rb}]$  of the target. Through experiments, it was found that  $[x, y, w, h]$  obtained better experimental results. In addition, as shown in Table 1 (5 and 6), we compared it with two other groups of experiments: the search image and the template image were concat and averaged, respectively. For Table 1 (5), all image features are fed directly into the DDC encoder. For Table 1 (6), the first step involves projecting them into a 1D embedding, followed by feeding them into the DDC encoder. From the experimental results, it can be observed that these two methods yield similar tracking performance.

**Integrating Online Update.** Our approach utilizes dynamic templates to capture the feature variations of the target object and select reliable templates. As shown in Table 1 (7), our method has improved tracking performance.

**Temporal Sequence.** We conducted an additional set of experiments showing the seamless integration of temporal information within our proposed framework. For instance, we constructed a time series that included the target’s coordinate positions in the previous 5 frames. We appended this time series before the START token. When generating the next new token, the historical frame coordinates were incorporated. Through this procedure, our method was capable of observing the target’s previous motion trajectory. Experimental results demonstrate that this integration approach enhances tracking performance across multiple datasets, as depicted in Table 1 (8).

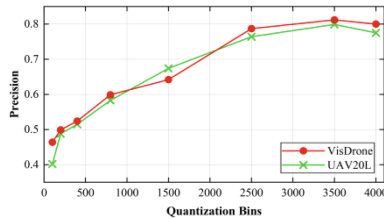


Fig. 7. Influence of the number of quantization bins.

**The Number of  $n_{bins}$ .** Additionally, we discussed the impact of the quantity of  $n_{bins}$  on the tracking, as shown in Fig. 7. What we analyzed is that the quantization error is reduced accordingly. As  $n_{bins}$  exceeds 3500, the performance gradually stabilizes, so we set it to 3500.

#### 4.4 Real-Time Analysis

**Results on UAV20L and DTB70.** As shown in Fig. 8, our tracker is compared with various SOTA tracking methods on the x-axis (FPS) and y-axis (Precision). DDCTrack demonstrates significantly superior performance in both speed and precision, outperforming several methods in terms of speed as well.

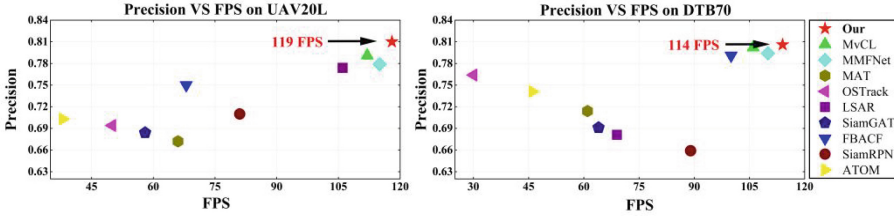


Fig. 8. Through quantitative comparisons on UAV20L [44] (left) and DTB70 [45] (right).

#### 4.5 Attribute-Based Comparison

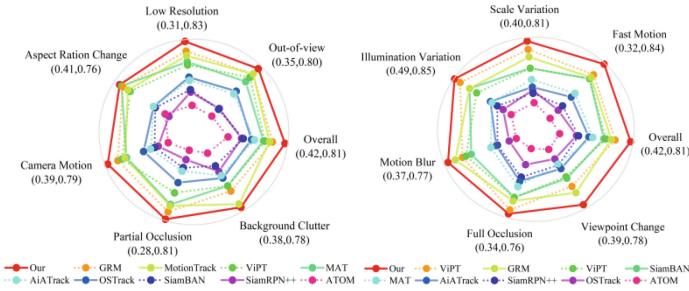


Fig. 9. The experimental results obtained on UAV in terms of overlap AUC with different challenging visual attributes.

As shown in Fig. 9, the experimental results demonstrate that DDCTrack outperforms current methods like GRM and ViPT when faced with these challenges. This is mainly attributed to our explicitly designed DDC module, which effectively learns global representations while dynamically adjusting the number of tokens. Additionally, our newly devised framework eliminates complex head networks, enhancing tracking performance in dealing with appearance variations.

## 5 Conclusion

Designing a simple, clean, and high-performance model for UAV tracking is a challenging task. In this work, we propose a novel dynamic token sampling for an efficient UAV transformer tracking framework, which addresses to some extent the drawbacks of the two-stream and two-stage models, such as complex head networks. Subsequently, based on the proposed DDC module, the framework dynamically selects tokens of significant information, allowing for the use of only the necessary tokens for each input video. This process discards some unnecessary tokens, significantly improving tracking speed. Experiments and analysis demonstrate the ability to achieve a good balance between performance and inference speed.

**Acknowledgements.** This work was supported by the China National Key Research and Development Program (2021YFB2802100) and the China National Science Foundation under Grant (62266044, 62061045, 61862061).

## References

1. Zhang, H., Li, Y., Liu, H., Yuan, D., Yang, Y.: Feature block-aware correlation filters for real-time UAV tracking. *IEEE Signal Process. Lett.* **31**, 840–844 (2024)
2. Zhong, Y., Shu, M.: Online background discriminative learning for satellite video object tracking. *IEEE Trans. Geosci. Remote Sens.* **62**, 1–15 (2024)
3. Gao, S., Zhou, C., Zhang, J.: Generalized relation modeling for transformer tracking. In: *Conference on Computer Vision and Pattern Recognition*, pp. 18686–18695 (2023)
4. Song, Z., Yu, J., Yang, W.: Transformer tracking with cyclic shifting window attention. In: *CVPR*, pp. 8781–8790 (2022)
5. Liang, Y., Ge, C., Tong, Z., Xie, P.: Not all patches are what you need: expediting vision transformers via token reorganizations. in: *ICLR*, arXiv preprint [arXiv:2202.07800](https://arxiv.org/abs/2202.07800) (2022)
6. Mu, H., Xia, W., Che, W.: Improving domain generalization for sound classification with sparse frequency-regularized transformer. In: *ICME*, pp. 1104–1108 (2023)
7. Dong, Y., Li, F., Ma, C., He, C., Wang, Z.J.: UAV-based dynamic object tracking with radio map. In: *ICASSP*, pp. 9166–9170. Seoul (2024)
8. Wang, W., et al.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: *ICCV*, pp. 568–578 (2021)
9. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: *ICLR* (2020)
10. Zhu, Z., Hou, J., Wu, D.: Cross-modal orthogonal high-rank augmentation for RGB-event transformer-trackers. In: *ICCV*, pp. 21988–21998 (2023)
11. Zheng, M., Li, H., Dong H.: End-to-end object detection with adaptive clustering transformer. arXiv preprint [arXiv:2011.09315](https://arxiv.org/abs/2011.09315) (2021)
12. Zeng, W., Jin, S., Liu, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: *CVPR*, pp. 11101–11111 (2022)
13. Yuan, D., et al.: Thermal Infrared target tracking: a comprehensive review. *IEEE Trans. Instrum. Meas.* **73**, 1–19 (2024)
14. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: *ICCV*, pp. 10448–10457 (2021)
15. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: *CVPR*, pp. 8126–8135 (2021)
16. Cha, J., Mun, J., Roh, B.: Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In: *CVPR*, pp. 11165–11174 (2023)
17. Alayrac, J.B., Donahue, J., Lenc, K., Mensch, A., Millican, K., Reynolds, M.: Flamingo: a visual language model for few-shot learning (2022)
18. Chu, X., Tian, Z., Xia, H., Shen, C.: Conditional positional encodings for vision transformers, arXiv preprint [arXiv:2102.10882](https://arxiv.org/abs/2102.10882) (2021)
19. Touvron, H., Cord, M., Jegou, H.: Training data-efficient image transformers and distillation through attention. *ICML*, arXiv preprint [arXiv:2012.12877](https://arxiv.org/abs/2012.12877) (2021)
20. Devlin, J., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding, arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
21. Shivwanshi, R., Nirala, N.: Implementation of an advanced lung nodule classification system using optimized ConvMixer and AdamW-based CNN architecture. *Signal Process. Algorithms, Archit. Arrangements, Appl.* **54–59** (2023)
22. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: SeqTrack: sequence to sequence learning for visual object tracking [arXiv:2304.14394](https://arxiv.org/abs/2304.14394) (2023)
23. Zhu, J., Lai, S., Chen, X., Wang, D., Lu, H.: Visual prompt multi-modal tracking. In: *Conference on Computer Vision and Pattern Recognition*, pp. 9516–9526 (2023)
24. Paul, M., Danelljan, M., Mayer, C., Gool, L.: Robust visual tracking by segmentation. In: *ECCV*, pp. 571–588 (2022)

25. Zhang, H., et al.: Feature block-aware correlation filters for real-time UAV tracking. *IEEE Signal Process. Lett.* **31**, 840–844 (2024)
26. Zhao, H., Wang, D., Lu, H.: Representation learning for visual object tracking by masked appearance transfer. In: *CVPR*, pp. 18696–18705 (2023)
27. Sun, D., et al.: UAV-ground visual tracking: a unified dataset and collaborative learning approach. *IEEE TCSVT* **34**, 3619–3632 (2024)
28. Liu, J., et al.: Online learning samples and adaptive recovery for robust RGB-T tracking. *IEEE TCSVT* **34**, 724–737 (2024)
29. Mayer, C., Danelljan, M., Paudel, D.P., Gool, L.: Learning target candidate association to keep track of what not to track. In: *ICCV*, pp. 13444–13454 (2021)
30. Li, Z., et al.: Material-guided multiview fusion network for hyperspectral object tracking. *IEEE TGRS* **62**, 1–15 (2024)
31. Ye, B., Chang, H., Ma, B., Shan, S.: Joint feature learning and relation modeling for tracking: a one-stream framework. In: *ECCV*, pp. 341–357 (2022)
32. Gao, S., Zhou, C., Ma, C., Wang, X., Yuan, J.: AiATrack: attention in attention for transformer visual tracking. In: *ECCV*, pp. 146–164 (2022)
33. Zhou, L., Zhou, Z., Mao, K., He, Z.: Joint visual grounding and tracking with natural language specification [arXiv:2303.12027](https://arxiv.org/abs/2303.12027) (2023)
34. Ma, F., et al.: Unified transformer tracker for object tracking. In: *CVPR*, pp. 8781–8790 (2022)



# HAPTICS: Human Action Prediction in Real-time via Pose Kinematics

Niaz Ahmad<sup>1</sup>, Saif Ullah<sup>1</sup>, Jawad Khan<sup>2</sup>, Chanyeok Choi<sup>1</sup>, and Youngmoon Lee<sup>1</sup>(✉)

<sup>1</sup> Hanyang University, Ansan, South Korea

{niazahmad89, fahad7878, angledssugar, youngmoonlee}@hanyang.ac.kr

<sup>2</sup> Gachon University, Seongnam, South Korea

jkhanbk1@gachon.ac.kr

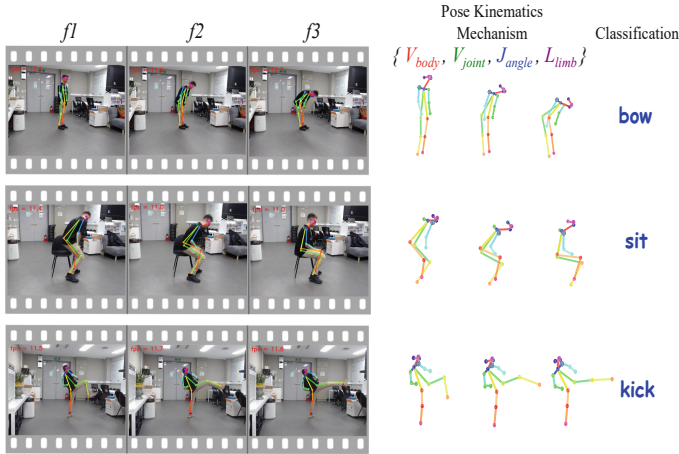
**Abstract.** Recognizing human actions in real-time presents a fundamental challenge, particularly when humans interact with other humans or objects in a shared space. Such systems must be able to recognize and assess real-world human actions from different angles and viewpoints. Consequently, a substantial volume of multi-dimensional human action training data is essential to enable data-driven algorithms to operate effectively in real-world scenarios. This paper introduces the Action Clip dataset, which provides a comprehensive 360-degree view of human actions, capturing rich features from multiple angles. Additionally, we describe the design and implementation of Human Action Prediction via Pose Kinematics (HAPTICS), a comprehensive pipeline for real-time human pose estimation and action recognition, all achievable with standard monocular camera sensors. HAPTICS utilizes a skeleton modality by transforming initially noisy human pose kinematic structures into skeletal features, such as body velocity, joint velocity, joint angles, and limb lengths derived from joint positions, followed by a classification layer. We have implemented and evaluated HAPTICS using four different datasets, demonstrating competitive state-of-the-art performance in pose-based action recognition and real-time performance at 30 frames per second on a live camera. The code and dataset are available at: <https://github.com/RaiseLab/HAPTics>

## 1 Introduction

Understanding human actions and behavior in a human-like manner presents a significant challenge for autonomous systems, robots, and their interactions with humans. In the context of self-driving cars and robots operating in urban environments [4][31], a future is envisioned where autonomous systems and humans coexist in shared public spaces. However, accurately inferring and predicting human actions in real-time using various sensor technologies remains a formidable task.

There are two primary challenges in human action recognition: (i) meeting real-time performance requirements, and (ii) acquiring suitable real-world training datasets. In scenarios where autonomous systems must engage with individuals, they require

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_10](https://doi.org/10.1007/978-3-031-78354-8_10).



**Fig. 1.** Action features obtained from a group of three frames  $f$  using pose kinematics mechanism denoted in colors, i.e., body velocity  $V_{body}$  (red), joint velocity  $V_{joint}$  (green), joint angles  $J_{angle}$  (blue), and length of limbs  $L_{limb}$  (violet) from joint positions. (Color figure online)

precise information about the specific actions people are undertaking in their immediate surroundings. This is especially critical in direct interactions with humans [10]. Given the highly dynamic nature of human actions, predicting them accurately and in real-time is paramount.

In addition to the runtime requirements of algorithms, data-driven approaches necessitate vast volumes of real-world training data. Data acquisition often emerges as a significant challenge in developing these algorithms, making the availability of ample data a critical aspect of algorithm design. Given the domain shift from heavily annotated visual data to real-world, synchronous, unlabeled data, pose-based action recognition algorithms must operate without direct reliance on visual annotations. Motivated by this, we recognized the substantial potential of leveraging real-time data with a 360-degree view of human actions for training our action recognition algorithms. Our objective is to facilitate the training of such algorithms using live real-world data, thereby overcoming domain transfer hurdles. This approach promises to significantly reduce the manual effort typically associated with recorded and annotated sensor data.

Several proposals for human action recognition [49] [47] [40] [50] acknowledge the challenges of real-time execution and real-world datasets and aim to develop datasets and efficient inference models. Recent studies suggest that skeletal-based action recognition [23] [49] [47], combined with CNN-based models [22] [26], and efficient networks [40] [50], can meet the latency requirements for training and inference in human action recognition. However, existing proposals suffer from either high computational complexity [23] [49] [47], notably inferior performance [40] [50], or sensitivity to variations in viewpoint [22] [26], rendering them impractical.

In this paper, we adopt a skeleton-based modality and present HAPTICS, a real-time human action predictor capable of capturing a 360-degree view of human actions in real-world settings. It performs real-time human pose estimation and action recogni-

tion using only a standard monocular camera. Specifically, building on a human pose extractor [5], HAPTICS maintains the receptive field while reducing computation and convolutional operations by replacing each  $7 \times 7$  convolutional kernel with three consecutive  $3 \times 3$  kernels. Additionally, the output of each of the three convolutional kernels is concatenated. The number of non-linearity layers is tripled, allowing the network to retain both lower and higher-level features. Finally, Part Affinity Fields (PAF) [5] are utilized to predict keypoint confidence maps and bipartite graph matching [44] is used to assemble the connections that share the same part detection candidates into full-body poses. Once the full-body pose task is performed, preprocessing operations are conducted to verify the actual kinematic structure of the human pose. Extensive skeletal feature extraction follows, including body velocity, joint velocity, joint angles, and limb lengths derived from joint positions to classify various human actions, as shown in Fig. 1.

We evaluated the HAPTICS system using four different action recognition benchmarks, including our proposed Action Clip dataset, NW-UCLA [41], NTU RGB+D 60 [35], and NTU RGB+D 120 [25]. To the best of our knowledge, HAPTICS is the first real-time model with reliable performance for the task of human pose estimation and action recognition, running at 30 frames per second (fps) on a live webcam using a single TITAN RTX. This paper makes the following contributions:

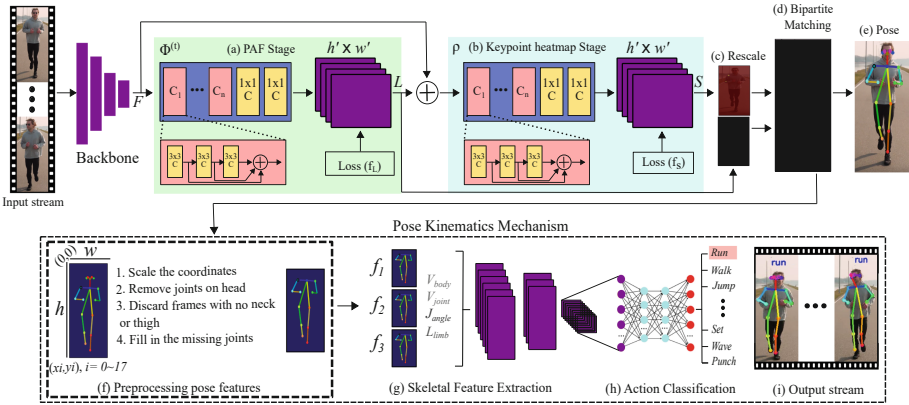
- Development of the Action Clip dataset, providing a  $360^\circ$  view of human actions to enhance the classifier’s ability to understand human action behaviors from various angles.
- Implementation of HAPTICS, a simple yet effective end-to-end pipeline with basic preprocessing and extensive skeletal feature extraction techniques.
- In-depth evaluation of HAPTICS regarding real-time execution and its effectiveness for real-world applications.

## 2 Related Work

**Skeletal-Based Action Recognition.** Action recognition based on skeletal data has received increasing attention due to its compactness compared to RGB-based representations. In a prior study [23], a framework for convolutional co-occurrence feature learning was introduced, employing a hierarchical approach to systematically integrate diverse levels of contextual information. The work by [49] proposes a view-adaptive model that autonomously adjusts observation perspectives during action occurrences, aiming to achieve view-invariant representations of human actions. However, CNN- or RNN-based models have played a significant role in this regard due to their substantial impact on spatial configurations.

Inspired by graph-based methods, Yan *et al.* [45] pioneered the integration of Graph Convolutional Networks (GCN) into skeleton-based action recognition, giving rise to ST-GCN. This model concurrently captures the spatial configurations and temporal dynamics of skeletons. Building upon this work, Song *et al.* [39] [38] addressed the occlusion issue in this domain by proposing a multi-stream GCN to extract rich features from more activated skeleton joints. Liu *et al.* [28] explored the impact of multi-adjacency GCNs and dilated temporal Convolutional Neural Networks (CNNs), intro-





**Fig. 2.** Overview of HAPTICS pipeline. (a/b) stages show architecture of the whole-body pose estimation pipeline, generate part affinity fields (PAFs)  $L$  and keypoint heatmaps  $S$  for torso, face, hand, and foot. The network is trained end-to-end with a multi-task loss  $(f_L)(f_S)$  combining each keypoint loss. (c) The most refined PAF and keypoint heatmap channels are resized at test time to improve accuracy. (d) The parsing algorithm utilizes the PAFs to identify whole-body parts belonging to the same person by performing bipartite matching. (e) The final whole-body poses are returned for all individuals in the frame. (f) The preprocessing stage operates on the whole body pose and performs important transformations. (g) Action features like  $V_{body}$ ,  $V_{joint}$ ,  $J_{angle}$ ,  $L_{limb}$  from joint positions are extracted from the previous three frames  $f_i$ . (h) These features are forwarded to the classification stage. (i) Finally, the action predictions are performed in the live output stream.

ducing a sophisticated model known as MS-G3D to disentangle multi-scale graph convolutions. Furthermore, a study [8] introduced a decoupled GCN method to enhance graph modeling capability without incurring additional computational overhead.

To enable global joint relationship modeling, Shi *et al.* [36] integrated the Non-local method [43] into a two-stream GCN model, named 2s-AGCN, resulting in a substantial enhancement in recognition accuracy. Similar to 2s-AGCN, the Dynamic GCN, proposed by Ye *et al.* [47], introduced a novel approach to model global dependencies, leading to outstanding accuracy in skeleton-based action recognition. While these methods have achieved remarkable performance, the escalating computational demands of multi-stream structures present challenges to their real-world applicability. Consequently, the quest to mitigate the complexity of GCN models remains a formidable task.

**CNN-Based Action Recognition.** Given the robust classification capabilities of convolutional neural networks (CNNs), several recent studies [22] [18] [21] [26] [42] have sought to convert skeleton sequences into 2D images and subsequently utilize CNNs for classification. In some instances [22], [21], the x, y, and z coordinates in 3D space are assigned to the three channels of an image, with frame IDs corresponding to different rows and joint IDs corresponding to different columns. The coordinate values are normalized within the range of 0-255 based on either dataset statistics [22] or sequence statistics [22], [21]. Alternatively, some studies [18] use relative positions between

joints and reference joints (e.g., left shoulder, right shoulder, left hip, and right hip) to generate multiple images. Other approaches [21] [42] use 2D projection maps from joint trajectories onto different orthogonal planes as images. One method [26] represents a 5D space (3D coordinates, time label, joint label) as a 2D coordinate space and a 3D color space, generating 10 images from different assignments of the 5D space. These 10 images are then fed into 10 ConvNets for classification, with the results from the 10 models fused for the final prediction.

While most of the aforementioned works focus on mapping a skeleton sequence to images, they overlook the challenge posed by view variations in the skeleton data. In contrast, our approach employs ConvNets to capture complex features from a 360-degree view of human 2D skeletons for multi-dimensional CNN-based action recognition.

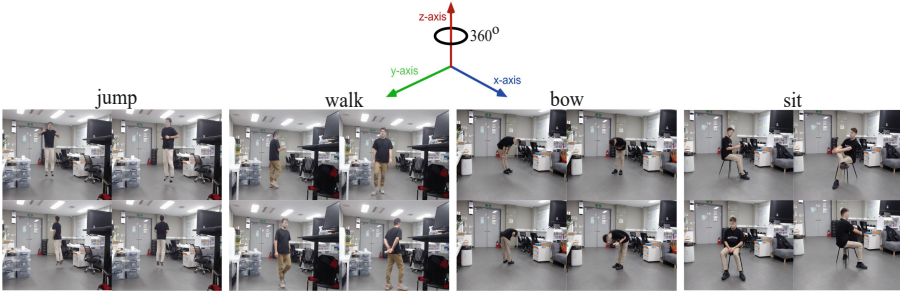
**Real-Time Models.** The efficiency of a model, commonly assessed by the number of trainable parameters and floating-point operations per second (FLOPs), is crucial in deep learning tasks. Numerous studies have focused on improving neural network efficiency, aiming to reduce model parameters or FLOPs. Examples include MobileNetV1 [14], MobileNetV2 [34], MobileNeXt [51], and EfficientNet [40]. The MobileNet family of models achieves size reduction through separable convolutions, which decompose standard convolutions into a depthwise convolution applied to each channel individually and a  $1 \times 1$  pointwise convolution to combine the outputs.

To refine structural hyperparameters in neural networks, compound scaling [40] introduces a family of EfficientNet models. In the context of skeletal-based action recognition, some studies have addressed the challenge of model complexity. For instance, the study in [46] devises a lightweight network with CNN-based blocks, though it lacks the accuracy of GCN models. Another study [50] employs a sophisticated data preprocessing strategy, incorporating positions, velocities, frame indexes, and joint types as inputs. While this preprocessing module enables the model to recognize actions with a shallow architecture, resulting in a rapid inference speed of 188 sequences per second per GPU, its performance is notably inferior to other state-of-the-art models.

## 3 Technical Approach

### 3.1 Action Clip Data Preparation

**Motivation.** The construction of the Action Clip dataset serves two primary purposes. First, it aims to establish a self-contained action dataset that excludes the involvement of a second object within the action (e.g., brushing teeth, where the toothbrush is the second object), as seen in [35] and [17]. This approach reduces the need for extensive manual image labeling for training, requiring only a video clip of any action performed by a human for feature extraction. Second, the dataset offers a generalized view with a 360-degree perspective of human actions in a live environment, enabling a comprehensive understanding of human activities from any angle. The decision to create the Action Clip dataset was motivated by the goal of improving the evaluation of the model’s performance across various views of human actions in real-time environments, ensuring compatibility with real-time applications.



**Fig. 3.** Examples of 360° view of the human body incorporated in Action Clip Dataset.

**Data Collection.** One notable advantage of our system is its ability to collect data from diverse sources. Our feature extraction algorithm can extract pose features from any human action video source, whether obtained through live video recording or downloaded from the internet. Initially, a continuous stream of video frames is transformed into still images, each labeled with a corresponding class (e.g., run, wave). In cases where duplicated images do not adequately describe the person’s action, these frames or a range of frames can be excluded. Subsequently, the video stream can span from a few seconds to minutes, capturing a specific type of action. Video clips are recorded with a frame size of  $640 \times 480$  and a frame rate of 10 fps, ensuring a rapid enough pace to capture the entire action movement. Visual examples of the 360-degree view of human actions<sup>1</sup> and the number of samples are illustrated in Fig. 3 and Table 1.

**Table 1.** The number of frames for each class used in training and testing operations.

Actions	stand	walk	run	jump	sit	squat	kick	punch	wave	bow	sleep	clap	total
# of frames	6196	4985	2042	4575	6881	2605	5105	5592	9805	5674	3628	5471	62559

### 3.2 Network Architecture

The pose pipeline is divided into two distinct stages, as shown in Fig. 2a and b, where refinement is applied to both the affinity field and confidence map branches at each stage. A significant advancement over [5] has been made to reduce computational costs. In our updated approach, refinement occurs exclusively at the PAF stage, where body part locations are already identified. As a result, confidence map prediction takes place solely at the PAF stage in Fig. 2. This adjustment reduces the computational load per stage by half. Empirical observations demonstrate that refined affinity field predictions enhance confidence map results at the keypoint heatmap stage, as shown in Fig. 2. Intuitively, the PAF channel identifies the locations of body parts, while the heatmap channel identifies the locations of keypoints.

<sup>1</sup> Refer to the supplementary materials for a detailed visual description.

Moreover, we increased the network depth. In the initial methodology proposed by [5], the network architecture featured multiple  $7 \times 7$  convolutional layers. In our updated model, we maintain the receptive field while reducing computation by replacing each  $7 \times 7$  convolutional kernel with three consecutive  $3 \times 3$  kernels. The former method required  $2 \times 7^2 - 1 = 97$  operations, while our current approach needs only 17. Additionally, we concatenate the output of each  $3 \times 3$  convolutional kernel, similar to DenseNet [15]. This triples the number of non-linearity layers, preserving both lower-level and higher-level features. Batch normalization is essential for the convergence of our deeper architecture; however, its implementation introduces a slowdown of approximately 20%. As an alternative, we replace ReLU layers with PReLU layers, which aid in fast convergence similar to batch normalization.

### 3.3 PAF-based Body Pose Estimation

The proposed pose estimation pipeline is based on the Part Affinity Field (PAF) architecture [5]. This methodology iteratively predicts Part Affinity Fields (PAFs) that encode part-to-part associations and detection confidence maps for keypoints. Each PAF is represented as a 2D orientation vector pointing from one keypoint to another. The input image is initially processed by a convolutional network (e.g., CMU Pose or Mobilenet-thin), generating a set of feature maps,  $F$ . Subsequently,  $F$  is fed into the PAF stage,  $\Phi^{(1)}$ , of the network  $\Phi$ , which predicts a set of PAFs,  $L^{(1)}$ . For each subsequent stage  $t$ , the PAFs from the previous stage,  $L^{(t-1)}$ , are concatenated with  $F$  and refined to produce  $L^{(t)}$ . After  $N$  stages, the final set of PAF channels is obtained as  $L = L^{(N)}$ . Finally,  $F$  and  $L$  are concatenated and fed into a network  $\rho$ , which predicts the keypoint heatmaps,  $S$ .

$$L^{(1)} = \Phi^{(1)}(F) \quad (1)$$

$$L^{(t)} = \Phi^{(t)}(F, L^{(i-1)}), \forall 2 \leq t \leq N \quad (2)$$

$$L = L^{(N)} \quad (3)$$

$$S = \rho(F, L) \quad (4)$$

L2 loss function is applied at each stage, comparing the estimated predictions with the groundtruth maps ( $S^*$ ) and fields ( $L^*$ ) for each pixel ( $p$ ) on each keypoint heatmap ( $c$ ) and PAF channel ( $f$ ):

$$f_L = \sum_{f=1}^F \sum_p (W(p) \cdot \|L_f(p) - L_f^*(p)\|_2^2) \quad (5)$$

$$f_S = \sum_{c=1}^C \sum_p (W(p) \cdot \|S_c(p) - S_c^*(p)\|_2^2) \quad (6)$$

Here,  $C$  and  $F$  represent the number of stages for predicting the keypoint heatmap and PAF, respectively. Additionally,  $W$  denotes a binary mask where  $W(p) = 0$  when an annotation is absent at pixel  $p$ . Non-maximum suppression is performed on the confidence map of keypoints to derive a discrete set of candidate locations for body parts.

Finally, bipartite graph matching [44] is employed to connect and assemble the detected parts into full-body poses for each individual in the frame, as shown in Fig. 2d and e.

**Table 2.** Notations in Algorithm 1.

$D_{raw}$	Raw skeleton data (joining positions)
$D_{nor}$	Normalize skeleton data
$D$	Action dataset
$F$	Frame in dataset
$D_{skeleton}$	Detected skeleton
$O_{miss}$	Missing joining position
$X_{ni\_curr}$	$X$ joint coordinate in current frame
$X_{ni\_prev}$	$X$ joint coordinate in previous frame
$Y_{mi\_curr}$	$Y$ joint coordinate in current frame
$Y_{mi\_prev}$	$Y$ joint coordinate in previous frame

**Table 3.** Notations in Algorithm 2.

$X_s$	Concatenation of joints pose of $f$ frames
$H$	Average skeleton height in $X_s$
$V_{body}$	Velocity of neck/H
$X$	Normalize pose
$V_{joints}$	Velocity of all joints in $X$

### 3.4 Preprocessing Pose Features

The initial skeleton data undergoes a preprocessing stage before feature extraction. This preprocessing comprises four distinct steps, as outlined in Fig. 2f preprocessing stage:

**Coordinate Scaling.** Initially, the keypoint positions obtained from our pose pipeline exhibit different units for the  $x$  and  $y$  coordinates. To ensure consistency and accommodate images with varying height and width ratios, these coordinates are uniformly scaled to a common unit:  $x', y' = x \cdot \text{scale\_factor}, y \cdot \text{scale\_factor}$ .

**Exclusion of Head Joints.** Our pose pipeline provides five keypoints related to the head, including the nose, two eyes, and two ears. However, for the specific actions within the training dataset, the positional information of the head minimally contributes to the classification task. The critical focus is on the body and limb configurations. Therefore, the five head joints are excluded to enhance the interpretability of features.

**Frame Discardance Criteria.** If a frame lacks a detected human skeleton or if the detected skeleton lacks neck or thigh joint information, the frame is considered invalid and is subsequently discarded. Additionally, the sliding window must be re-initialized on the next valid frame in such cases.

**Handling Missing Joints.** In certain scenarios, the pose estimation pipeline may fail to detect a complete human skeleton within an image, resulting in gaps or missing joint positions. To maintain a fixed-size feature vector for subsequent classification, these missing joints must be assigned values. Two suboptimal solutions were considered: (1) rejecting the frame, which was impractical as it would prevent action detection when individuals were oriented sideways or not facing the camera, and (2) assigning positions outside of a reasonable range, which could theoretically work with a robust classifier. However, empirical results showed poor recognition accuracy with this method, making it unsuitable. Instead, a more effective solution was adopted, involving the assignment of missing joint positions based on their relative positions in the preceding frame concerning the neck joint. For example, if in the previous frame, the hand was located 10

pixels to the right of the neck, and in the current frame, the hand is missing, it is placed 10 pixels to the right of the neck in the current frame. Empirical experimentation confirmed the effectiveness of this approach. Step-by-step pseudocode for preprocessing pose features is presented in Algorithm 1. Table 2 summarizes the notations used in Algorithm 1.

---

**Algorithm 1** Preprocessing Pose Features
 

---

**Input:**  $D_{raw} = \{(X_{a1}, Y_{b1}), (X_{a2}, Y_{b2}), \dots, (X_{a17}, Y_{b17})\}$   
**Output:**  $D_{nor} = \{(X_{n1}, Y_{m1}), (X_{n2}, Y_{m2}), \dots, (X_{n17}, Y_{m17})\}$

**Step 1:** Scale the coordinates  
**for**  $(X_{ai}, Y_{bi}) \in D_{raw}$  **do:**  
      $D_{nor} = \text{Normalize} \{(X_{ai}, Y_{bi})\}$   
**end for**

**Step 2:** Remove head joint  
**for**  $(X_{ni}, Y_{mi}) \in D_{nor}$  **do:**  
     **if**  $(X_{ni}, Y_{mi}) = \{'head', 'eyes', 'ears'\}$  **then**  
          $\text{Discard}(X_{ni}, Y_{mi})$   
     **end if**  
**end for**

**Step 3:** Discard frames with no neck or thigh  
**for**  $F \in D$  **do:**  
     **if**  $D_{skeleton} = \emptyset$  **then:**  
          $\text{Discard } F$   
     **end if**  
**end for**

**Step 4:** Fill in the missing joints  
**for**  $D_{skeleton} \in F$  **do:**  
     **if**  $O_{miss} \in F$  **then:**  
          $X_{ni\_curr} = X_{Neck_{curr}} + (X_{ni\_prev} - X_{Neck_{prev}})$   
          $Y_{mi\_curr} = Y_{Neck_{curr}} + (Y_{mi\_prev} - Y_{Neck_{prev}})$   
     **end if**  
**end for**

---

### 3.5 Skeletal Feature Extraction

Following the initial preprocessing step, the joint positions are now complete and ready for further analysis, as shown in Fig. 2g. In this section, we utilize the joint positions obtained from a sequence of  $f = 3$  frames as raw features. Additionally, we design and extract distinctive features to enhance the discrimination of various types of actions. A step-by-step overview of the computed features is presented in Algorithm 2, with details of the notations provided in Table 3.

Before initializing model training, the normalized pose data—namely body velocity ( $V_{body}$ ), joint velocity ( $V_{joint}$ ), joint angles ( $J_{angle}$ ), and limb lengths ( $L_{limb}$ )—are selected as trainable features. These features are concatenated to create a feature vector with a dimensionality of 170. Subsequently, the Principal Component Analysis (PCA) algorithm is employed to reduce the dimensionality of the feature vector. After applying PCA, the dimensionality is reduced by 70%, aiming to decrease training time and computational costs. These meticulously engineered features are now prepared for training the classifier, as depicted in Fig. 2h.

## 4 Evaluation

### 4.1 Datasets and Experimental Setup

We evaluate the proposed HAPTICS model on four challenging benchmarks: NTU RGB+D 60 [35], NTU RGB+D 120 [25], Northwestern-UCLA [41], and our proposed Action Clip dataset. The evaluation protocols for NTU RGB+D 60, NTU RGB+D 120, and Northwestern-UCLA follow those outlined in their respective published papers.

All experiments use a single TITAN RTX GPU under the PyTorch deep learning framework. The models are trained using Stochastic Gradient Descent with a momentum of 0.9 and a weight decay of 0.0004. Pose features are extracted from a sequence of 3 frames using CMU pose [5] and Mobilenet-thin [14] backbone networks. The end-to-end training is performed using an input image size of  $656 \times 368$ , and the same resolution is consistently maintained throughout the experiments.

---

#### Algorithm 2 Skeletal Feature Extraction

---

```

Step 1: Calculate  $X_s$ 
Initialize  $X_s$  as a dynamic array
for  $i = 1$  to 3 do:                                     ▷ f = 3 frames
  for  $j = 1$  to 13 do:                                   ▷ 13 joints
    for  $k = 1$  to 2 do:                                   ▷ 2 position values per joint
       $X_s[(i - 1) * 26 + (j - 1) * 2 + k] =$  joint position ( $i, j, k$ )
    end for
  end for
end for
Step 2: Calculate  $H$ 
Initialize  $H$ 
for  $i = 1$  to 3 do:                                     ▷ f = 3 frames
   $H = H +$  distance between neck position( $i$ ), thigh
  position( $i$ )
   $H = H/5$ 
end for
Step 3: Calculate  $V_{body}$ 
Initialize  $V_{body}$  as a dynamic array
for  $i = 1$  to 3 do:                                     ▷ f = 3 frames
   $V_{body}[i - 2] =$  velocity(neck position( $i$ ), neck position
  ( $i - 1$ ))/ $H$ 
end for
Step 4: Calculate  $X$ 
Initialize  $X$  as a dynamic array
 $X_{mean} = \text{mean}(X_s)$ 
for  $i = 1$  to 78 do:                                     ▷ joint positions in 3 frames
   $X[i] = (X_s[i] - X_{mean})/H$ 
end for
Step 5: Calculate  $V_{joints}$ 
Initialize  $V_{joints}$  as a dynamic array
for  $i = 1$  to 3 do:                                     ▷ f = 3 frames
  for  $j = 1$  to 13 do:                                   ▷ 13 joints
    for  $k = 1$  to 2 do:                                   ▷ 2 velocity values per joint
       $V_{joints}[(i - 1) * 23 + (j - 1) * 2 + k] =$  joint velocity ( $i, j, k$ )
    end for
  end for
end for
Step 5: Calculate Joint angles from  $X_s$ 
Step 6: Calculate the length of each limb from  $X_s$ 

```

---

## 4.2 Results

**Comparison with State-of-the-Art.** We initially evaluated the proposed system on our newly launched Action Clip dataset, which includes 12 challenging actions captured from a  $360^\circ$  view of daily life activities. Each action was assessed by calculating precision, recall, and F1 score using test samples from the dataset, as shown in Table 4. Our system achieved 97% accuracy across all three metrics. Additionally, we imple-

**Table 4.** Precision, recall, and f1-score on our proposed Action Clip test dataset. † indicates CMU pose used as a human pose feature extractor.

Action	Precision	Recall	F1-score	Test set
jump	0.94	0.95	0.95	1352
kick	0.97	0.97	0.97	1525
punch	0.97	0.96	0.97	1709
run	0.97	0.95	0.96	619
sit	1.00	0.99	0.99	2008
squat	0.99	0.97	0.98	737
stand	0.96	0.97	0.97	1795
walk	0.94	0.94	0.94	1448
wave	0.99	0.99	0.99	2951
bow	0.94	0.97	0.97	1747
sleep	0.99	1.00	1.00	1092
clap	0.96	0.97	0.97	1720
Accuracy †			0.97	18702
Macro avg †	0.97	0.97	0.97	18702
Weight avg †	0.97	0.97	0.97	18702

**Table 6.** Comparisons with the state-of-the-art methods on NTU RGB+D 60 dataset. † indicates Mobilenet-thin and ‡ indicates CMU pose used as a human pose feature extractor.

Methods	Modalities	X-sub	X-set
PoseConv3D [11]	RGB	93.7	96.6
ActionMachine [52]	RGB	94.3	97.2
Glimpse Clouds [3]	RGB	86.6	93.2
SRNet [30]	Skeleton	87.3	91.3
AGC-LSTM [37]	Skeleton	89.2	95.0
Shift-GCN [9]	Skeleton	90.7	96.5
TemPose [16]	Skeleton	92.7	95.2
HAPTICS †	Skeleton	96.8	93.5
HAPTICS ‡	Skeleton	97.3	94.2

**Table 5.** Comparison of the Top-1 accuracy (%) with the state-of-the-art methods on our proposed Action Clip dataset. Results are implemented based on their released codes. † indicates Mobilenet-thin and ‡ indicates CMU pose used as a human pose feature extractor.

Methods	Modalities	Accuracy
nCTE [13]	RGB	66.7
Glimpse Clouds [3]	RGB	90.8
ActionMachine [52]	RGB	95.3
TS-LSTM [20]	Skeleton	87.9
Shift-GCN [9]	Skeleton	93.2
CTR-GCN [6]	Skeleton	95.4
HAPTICS †	Skeleton	95.8
HAPTICS ‡	Skeleton	97.0

**Table 7.** Comparison with the state-of-the-art methods on NTU RGB+D 120 dataset. † indicates Mobilenet-thin and ‡ indicates CMU pose used as a human pose feature extractor.

Methods	Modalities	X-sub	X-set
PoseConv3D [11]	RGB	86.0	89.6
Shift-GCN [9]	Skeleton	85.9	87.6
KA-AGTN [27]	Skeleton	86.1	88.0
TemPose [16]	Skeleton	87.0	88.5
4s-MTS-Former [29]	Skeleton	87.1	90.0
MSSTNet [12]	Skeleton	87.4	88.4
CTR-GCN [6]	Skeleton	88.9	90.6
HAPTICS †	Skeleton	86.1	88.3
HAPTICS ‡	Skeleton	89.7	91.5

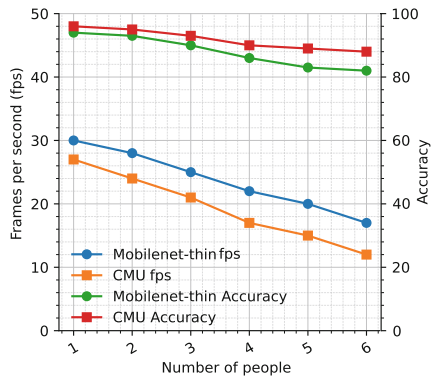


mented systems using both RGB and skeleton modalities with the Action Clip dataset and compared the results with our proposed system, as presented in Table 5.

We also evaluated the proposed model on the NTU RGB+D 60 and NTU RGB+D 120 datasets, following the X-sub and X-set protocols. The results, presented in Tables 6 and 7, demonstrate the effectiveness of our model. HAPTICS achieved the highest accuracy of 97.3% on X-sub and 94.2% on X-set for the NTU RGB+D 60 benchmark, surpassing state-of-the-art methods [30], [37], [9], and [16]. Similarly, our proposed technique achieved the highest accuracy of 89.7% on X-sub and 91.5% on X-set for the NTU RGB+D 120 benchmark, marking significant improvements over state-of-the-art methods [16], [29], [12], and [6], respectively.

**Table 8.** Comparisons of top-1(%) accuracy with state-of-the-art methods on the Northwestern-UCLA dataset. HAPTICS uses Mobilenet-thin ( $\dagger$ ) and CMU pose ( $\ddagger$ ) networks for human pose estimation.

Methods	Modalities	Top-1
NKTM [33]	RGB	75.8
Glimpse Clouds [3]	RGB	91.1
Action Machin [52]	RGB	96.5
TS-LSTM [20]	Skeleton	89.2
AGC-LSTM [37]	Skeleton	93.3
Shift-GCN [9]	Skeleton	94.6
CTR-GCN [6]	Skeleton	96.5
MSSTNet [12]	Skeleton	97.6
HAPTICS $\dagger$	Skeleton	97.4
HAPTICS $\ddagger$	Skeleton	98.2



**Fig. 4.** Frames per second and accuracy for the given number of people in live webcam using TITAN RTX GPU.

Finally, we evaluated our proposed model on the low-resolution Northwestern-UCLA dataset to validate its effectiveness and generalizability. Table 8 presents promising results compared to top competitors [20], [37], [9], [6], and [12]. Specifically, HAPTICS achieved a top-1 accuracy of 98.2%, representing a 3.6% improvement over [9], a 1.7% improvement over [6], and a 0.6% improvement over the recent [12].

**Computational Speed and Cost.** Computational speed is measured based on the execution time in frames per second (fps) using live webcam video at a resolution of  $656 \times 368$ . Fig. 4 shows that HAPTICS, with Mobilenet-thin, achieves the highest fps, reaching 30 for a single person and approximately 17 fps when the number of individuals increases to 6. In contrast, the CMU method attains high accuracy but has slightly lower fps due to the nature of its design. Fig. 5 presents the computation cost in FLOPS, providing a comparison to methods that use skeleton modalities.

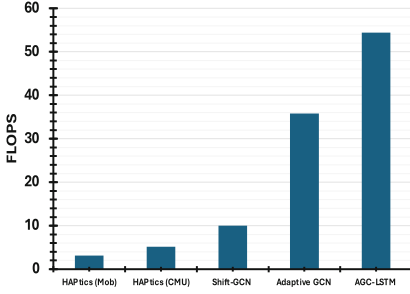


Fig. 5. Computational cost compared with skeleton modalities systems.

Table 9. Pose estimation comparison using COCO keypoint *test* set.

Models	Backbone	AP	AP <sup>M</sup>	AP <sup>L</sup>
PersonLab [32]	ResNet-152	68.7	64.1	75.5
MultiPoseNet [19]	ResNet-101	69.6	65.0	76.3
HigherHRNet [7]	HRNet	70.5	66.6	75.8
SIMPLE [48]	HRNet-W32	71.1	69.1	79.1
StrongPose [2]	ResNet-101	72.1	67.0	77.1
PosePlusSeg [1]	ResNet-152	72.8	67.8	79.4
<b>HAPTICS</b>	<b>Mobilenet</b>	74.6	69.1	81.7
<b>HAPTICS</b>	<b>CMU</b>	76.3	71.7	83.8

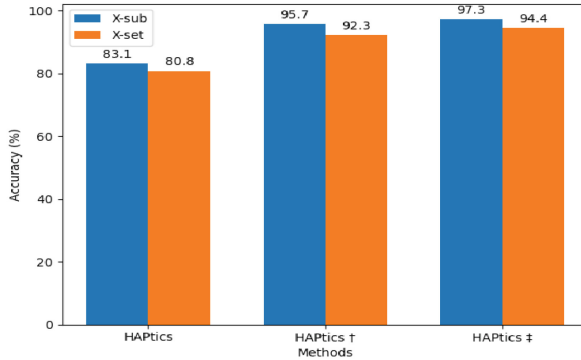
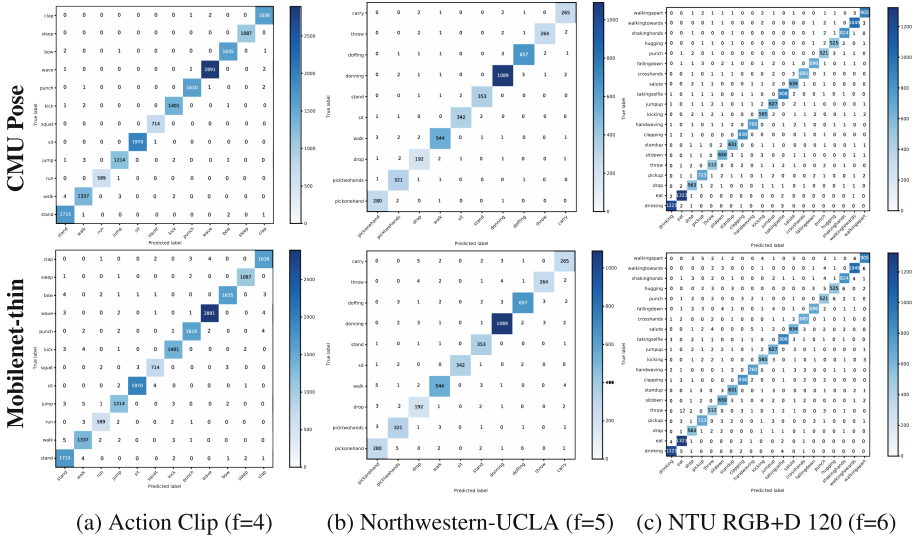


Fig. 6. HAPTics represents the results of the baseline version of our model that uses only the CMU pose backbone network. HAPTics † denotes results obtained when the functionality of PAF is enabled, while HAPTics ‡ indicates the results achieved with the integration of PAF and advanced feature extraction.

### 4.3 Ablation Studies

**Adaptation of PAF.** We evaluated the generalizability of our pose pipeline by integrating the PAF technique [5]. Our findings outperforms recent state-of-the-art methods, including SIMPLE [48], StrongPose [2], and PosePlusSeg [1], when evaluated on the COCO keypoint test set [24], as shown in Table 9.

**Backbone vs. Techniques.** To systematically assess the impact of our specific techniques beyond the capabilities of the backbone networks, we conducted studies comparing the system’s performance with and without the integration of PAF and our enhanced feature extraction techniques. The ablation results in Fig. 6 demonstrate that while the backbone networks establish a high baseline of performance, the integration of our methods provides significant additional improvements in accuracy. For instance, with PAF-based enhancements and advanced feature extraction methods, the system’s accuracy noticeably increased, particularly in complex action recognition scenarios.



**Fig. 7.** The confusion matrix illustrates the outcomes of 12 actions from (a) the Action Clip dataset, employing frames ( $f = 4$ ) for pose feature extraction, 10 action results from (b) the Northwestern-UCLA dataset, utilizing  $f = 5$  frames for pose feature extraction, and 20 novel fine-grained action classes from (c) the NTU RGB+D 120 dataset, with  $f = 6$  for pose feature extraction. (Best viewed with zoom in).

**Pose Features with Varied Numbers of Frames.** We explored the performance of HAPTICS across different numbers of frames ( $f = 4, 5, 6$ ) for pose feature extraction. Our investigation covered 12 classes from the proposed Action Clip Dataset, as depicted in Fig. 7(a), 10 classes from the NW-UCLA dataset, as illustrated in Fig. 7(b), and 20 novel classes from the NTU RGB+D 120 dataset, as shown in Fig. 7(c).

## 5 Conclusion and Future Work

This research proposes a novel Action Clip dataset that captures a  $360^\circ$  view of human actions and introduces a comprehensive pipeline for real-time human pose estimation and action recognition using standard monocular camera sensors. The proposed approach transforms noisy human pose kinematic features into encoded skeletal features. These features are then classified using deep neural network techniques, achieving not only competitive state-of-the-art performance in pose-based action detection but also ensuring real-time execution.

**Acknowledgments.** This work was supported in part by the National Research Foundation of Korea (NRF) grant 2022R1G1A1003531, 2022R1A4A3018824 and Institute of Information and Communications Technology Planning and Evaluation (IITP) grant RS-2020-II201741, RS-2022-00155885, RS-2024-00423071 funded by the Korea government (MSIT).

## References



1. Ahmad, N., Khan, J., Kim, J.Y., Lee, Y.: Joint Human Pose Estimation and Instance Segmentation with PosePlusSeg. In: AAAI (2022)
2. Ahmad, N., Yoon, J.: Strongpose: bottom-up and strong keypoint heat map based pose estimation. In: ICPR. pp. 8608–8615. IEEE (2021)
3. Baradel, F., Wolf, C., Mille, J., Taylor, G.W.: Glimpse clouds: Human activity recognition from unstructured feature points. In: CVPR. pp. 469–478 (2018)
4. Brooks, R.: The big problem with self-driving cars is people. *IEEE spectrum: technology, engineering, and science News* 27(8) (2017)
5. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. pp. 7291–7299 (2017)
6. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: ICCV. pp. 13359–13368 (2021)
7. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhnet: Scale-aware representation learning for bottom-up human pose estimation. In: CVPR (2020)
8. Cheng, K., Zhang, Y., Cao, C., Shi, L., Cheng, J., Lu, H.: Decoupling gcn with dropgraph module for skeleton-based action recognition. In: ECCV. pp. 536–553. Springer (2020)
9. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: CVPR. pp. 183–192 (2020)
10. Choi, C., Kim, J., Nam, Y.: Snapbot : Enabling Dynamic Human Robot Interactions for Real-Time Computational Photography. In: HRI (2024)
11. Duan, H., Zhao, Y., Chen, K., Lin, D., Dai, B.: Revisiting skeleton-based action recognition. In: CVPR. pp. 2969–2978 (2022)
12. Feng, D., Wu, Z., Zhang, J., Ren, T.: Multi-scale spatial temporal graph neural network for skeleton-based action recognition. *IEEE Access* 9, 58256–58265 (2021)
13. Gupta, A., Martinez, J., Little, J.J., Woodham, R.J.: 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In: CVPR (2014)
14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
15. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
16. Ibh, M., et al.: Tempose: a new skeleton-based transformer model designed for fine-grained motion recognition in badminton. In: CVPR. pp. 5199–5208 (2023)
17. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint [arXiv:1705.06950](https://arxiv.org/abs/1705.06950) (2017)
18. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: ICPR. pp. 3288–3297 (2017)
19. Kocabas, M., Karagoz, S., Akbas, E.: Multiposenet: Fast multi-person pose estimation using pose residual network. In: ECCV. pp. 417–433 (2018)
20. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In: CVPR. pp. 1012–1020 (2017)
21. Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y., He, M.: Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In: ICMEW. pp. 601–604 (2017)
22. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: ICMEW. pp. 597–600 (2017)

23. Li, C., Zhong, Q., Xie, D., Pu, S.: Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv preprint [arXiv:1804.06055](https://arxiv.org/abs/1804.06055) (2018)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. TPAMI **42**(10), 2684–2701 (2019)
26. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn. **68**, 346–362 (2017)
27. Liu, Y., Zhang, H., Xu, D., He, K.: Graph transformer network with temporal kernel attention for skeleton-based action recognition. Knowl.-Based Syst. **240**, 108146 (2022)
28. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: CVPR. pp. 143–152 (2020)
29. Lv, J., Gong, X.: Multi-grained temporal segmentation attention modeling for skeleton-based action recognition. IEEE Signal Processing Letters (2023)
30. Nie, W., Wang, W., Huang, X.: Srnet: Structured relevance feature learning network from skeleton data for human action recognition. IEEE Access **7**, 132161–132172 (2019)
31. Nwankwo, L., Rueckert, E.: The conversation is the command: Interacting with real-world autonomous robot through natural language. arXiv preprint [arXiv:2401.11838](https://arxiv.org/abs/2401.11838) (2024)
32. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In: ECCV. pp. 269–286 (2018)
33. Rahmani, H., Mian, A.: Learning a non-linear knowledge transfer model for cross-view action recognition. In: CVPR. pp. 2458–2466 (2015)
34. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR. pp. 4510–4520 (2018)
35. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: CVPR. pp. 1010–1019 (2016)
36. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: CVPR. pp. 12026–12035 (2019)
37. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: CVPR. pp. 1227–1236 (2019)
38. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Richly activated graph convolutional network for robust skeleton-based action recognition. Transactions on Circuits and Systems for Video Technology **31**(5), 1915–1925 (2020)
39. Song, Y.F., Zhang, Z., Wang, L.: Richly activated graph convolutional network for action recognition with incomplete skeletons. In: ICIP. pp. 1–5. IEEE (2019)
40. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML. pp. 6105–6114. PMLR (2019)
41. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.C.: Cross-view action modeling, learning and recognition. In: CVPR. pp. 2649–2656 (2014)
42. Wang, P., Li, W., Li, C., Hou, Y.: Action recognition based on joint trajectory maps with convolutional neural networks. Knowl.-Based Syst. **158**, 43–53 (2018)
43. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
44. West, D.B., et al.: Introduction to graph theory, vol. 2. Prentice hall (2001)
45. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI. vol. 32 (2018)
46. Yang, F., Wu, Y., Sakti, S., Nakamura, S.: Make skeleton-based action recognition model smaller, faster and better. In: MM, pp. 1–6 (2019)
47. Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H.: Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In: MM. pp. 55–63 (2020)

48. Zhang, J., Zhu, Z., Lu, J., Huang, J., Huang, G., Zhou, J.: Simple: Single-network with mimicking and point learning for bottom-up human pose estimation. arXiv preprint [arXiv:2104.02486](https://arxiv.org/abs/2104.02486) (2021)
49. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive neural networks for high performance skeleton-based human action recognition. *TPAMI* **41**(8) (2019)
50. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: *CVPR*. pp. 1112–1121 (2020)
51. Zhou, D., Hou, Q., Chen, Y., Feng, J., Yan, S.: Rethinking bottleneck structure for efficient mobile network design. In: *ECCV*. pp. 680–697 (2020)
52. Zhu, J., Zou, W., Xu, L., Hu, Y., Zhu, Z., Chang, M., Huang, J., Huang, G., Du, D.: Action machine: Rethinking action recognition in trimmed videos. arXiv preprint (2018)



# Predicting the Next Action by Modeling the Abstract Goal

Debaditya Roy<sup>1</sup> and Basura Fernando<sup>1,2</sup>

<sup>1</sup> Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore, Singapore

roy\_debaditya@ihpc.a-star.edu.sg, fernando\_basura@cfar.a-star.edu.sg

<sup>2</sup> Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore, Singapore

**Abstract.** The problem of predicting human actions from observed videos is an inherently uncertain one. We present an action anticipation model that leverages latent goal information to reduce the uncertainty in future predictions. We develop a latent variable representing goal information called abstract goal which is conditioned on observed sequences of visual features for action anticipation. We design the abstract goal as a distribution whose parameters are estimated using a variational recurrent model. We sample multiple candidates for the next action and use goal consistency criterion to determine the best candidate that follows from the abstract goal. Our method obtains impressive results on the very challenging Epic-Kitchens55 (EK55) and good results in Epic-Kitchens100 (EK100) datasets. Code is available at [https://github.com/LAHProject/Abstract\\_Goal](https://github.com/LAHProject/Abstract_Goal)

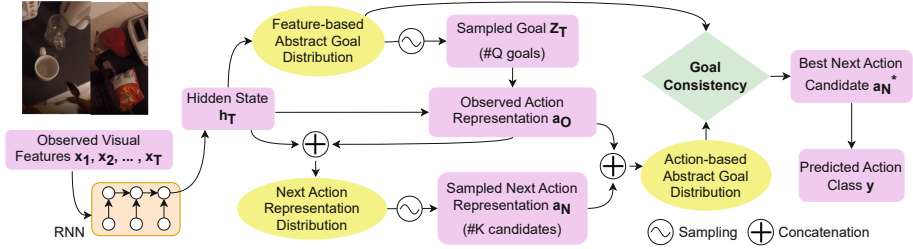
**Keywords:** Action Anticipation · Stochastic Modeling · Variational Inference

## 1 Introduction

Anticipating human actions from videos has significant relevance across various domains, including but not limited to human-robot collaboration, intelligent domiciles, assistive robotics, and wearable virtual assistants. Specifically, ego-centric videos, which capture the actions of the individual wearing the camera, represent a valuable resource for the development of intelligent assistants capable of forecasting the wearer's future actions and providing tailored assistance accordingly. A fundamental challenge in action anticipation lies in the inherent uncertainty surrounding future predictions. Human behavior is predominantly steered by individual goals or intentions, thus guiding the sequence of actions

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_11](https://doi.org/10.1007/978-3-031-78354-8_11).



**Fig. 1.** Model design for abstract goal-based action anticipation. Yellow ellipses represent distributions and pink boxes represent various variables of the model.

performed. Consequently, incorporating goal information holds promise for mitigating such uncertainty in forecasting future actions. For example, with information about the goal *wash pan*, a model can predict that *take pan* will be followed by *rinse pan* and not *put pan on stove*.

Goal and intentions have been adopted in some recent works for effective action anticipation [21, 27, 30]. In this paper, we make use of a stochastic method [4, 9] for latent goal modeling to improve action anticipation that goes beyond the deterministic latent goal representation in [27]. We propose to learn a new latent variable called abstract goal as a latent distribution as shown in Figure 1. We use two types of abstract goal distributions when predicting the next action in the sequence. The first abstract goal distribution is learned using the observed visual features and a stochastic recurrent neural network [4] which we call “feature-based abstract goal” distribution. Furthermore, we design an “action-based abstract goal” distribution using the next action representation distribution and the observed action representation. We sample multiple next-action-representation candidates and use the goal consistency criterion to find the most likely next action—see Figure 1. The action that is most likely to happen in the future (“next best action”) is the one that maximizes consistency between the two latent abstract goal distributions. During learning, we use goal consistency as a loss function to obtain a model informed of human behavior, i.e. the sequences of actions. Such a mechanism is not present in previous stochastic approaches [1, 21, 22] which only minimize KL divergence between prior and posterior latent distribution to obtain the best future actions. Also, we introduce a goal consistency measure to choose the best next action candidate rather than mean or median sampling used in [1, 22]. We show that goal consistency has the biggest impact on action anticipation. Our approach yields improvements when predicting the next action in unscripted activities on the Epic-Kitchens55 (EK55). Our contributions are:

- A new latent variable called abstract goal using a stochastic recurrent model that uses two latent distributions for the observed and the next action and enforces consistency among them to effectively predict the next action.
- A novel goal consistency term that measures how well a plausible future action (next action) aligns with abstract goal distributions.



## 2 Related work

Research in action anticipation has gained popularity in recent years thanks to progress in datasets [6] and challenges [5]. The activity label of the entire action sequence is used to anticipate the next action in [29]. In [27], observed features are used to obtain a fixed latent goal from visual features. [3] conceptualizes goals as the visual outputs of a sequence of actions. They predict each action in the sequence based on its relative closeness to the goal as compared to the previous action. [19] propose to use an external memory bank to store prototypes of the overall activity and contrastive learning augmented with the memory bank for forecasting the next action.

**Predicting Future features for Anticipation.** In [10] authors show that LSTM can be unrolled for multiple time steps to predict future features can be used to accurately predict the next action. In [26], Human-object interactions are encoded as features and fed to a transformer encoder-decoder to predict the features of future frames and the corresponding future actions. Authors in [18] estimate spatial attention maps of future human-object interactions to predict the next action. In [32], authors propose to summarize long-range sequences by processing smaller temporal sequences and caching them in memory as context and using the context for action anticipation. In [12], a real-time action anticipation framework is presented using a two-stage transformer with reduced parameters that is trained for future feature prediction and action anticipation. Due to the lightweight nature of their model, the action inference is performed in real-time. In [16], temporal features are computed using time-conditioned skip connections to anticipate the next action. In [33], an RNN is used to generate the intermediate frames between the observed frames and the anticipated action. In [13], every frame is represented using a Visual Transformer (ViT) [7] and combined using a temporal transformer to predict future features and action labels. Authors in [34], train a transformer model to predict the next action by reducing the amount of observed future available during learning from fully available to completely absent. Authors in [28] model interactions using cross-attention between humans and object visual features using a spatio-temporal visual transformer and use the modeled interaction to predict the next action.

**Long-term forecasting.** In [2], future actions and their duration are predicted autoregressively using an RNN with observed action labels as input. In [2, 20], RNNs are used to predict future actions conditioned on observed action labels. Latent distributions are used in literature to encode the observed action and duration in [1, 22]. In [1], a sample from the latent distribution of observed action is combined with previously predicted action in a decoder to predict the multiple next actions and their duration. In [22], two decoders are used to predict the action labels and duration separately. The action decoder uses the action labels in the observed video as input while the duration prediction decoder uses the duration of actions. Similarly, in [14], a transformer is used to encode past actions and duration while another transformer decoder is used to predict both future actions and their duration. In [24], authors use two transformer encoders for segment-level and long-term encoding and a decoder that fuses both encoder

inputs to predict future actions. In [21], goal labels and observed features are used as input to a conditional variational encoder to predict future actions. In [37], a large language model is prompted with observed actions and narrations to predict future actions.

**Correlating past and future.** In [23], authors model the transition between the visual features of the observed and the next action to generate the next action features. A similar action anticipation model that correlates past observed features with the future using Jaccard vector similarity is presented in [8]. In [16], time-conditioned skip connections are used to generate features for predicting future actions at different anticipation time in the future. In [11], authors propose a neural memory network to compare an input (spatial representation or labels) with the existing memory content to predict future action labels. Similarly, in [25], authors propose an action anticipation framework with a self-regulated learning process. A counterfactual reasoning is used to improve action anticipation in [36]. Our approach correlates the past and future by enforcing goal consistency between the two abstract goal distributions computed using observed features and the next action.

### 3 Action anticipation with abstract goals

In this section, we explain our model design outlined in Figure 1. At first, we explain how to compute the feature-based abstract goal distribution in Section 3.1. Then, we describe how to obtain next action candidates and action-based abstract goal with respect to these candidates in Section 3.2 and 3.3, respectively. We then explain the goal consistency criterion used to obtain the best next action candidate in Section 3.4. Finally, we describe the various loss functions to train our model in Section 3.5.

#### 3.1 Observed Feature-based abstract goal representation

In this section, we describe how to generate *feature-based abstract goal* representation using variational recurrent neural network (VRNN) framework [4, 9]. Let us denote the observed feature sequence by  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$  where  $\mathbf{x}_t \in \mathbb{R}^{d_f}$ . Following standard VRNN, a Gaussian distribution  $q_t(\mathbf{z}_t | \mathbf{x}_{1:t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior})$  is used to model the prior distribution of the abstract goal ( $\mathbf{z}_t$ ) given the observed feature sequence  $\mathbf{x}_{1:t-1}$ . The parameters  $\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior} \in \mathbb{R}^{d_z}$  are estimated using the hidden state of the RNN ( $\mathbf{h}_{t-1} \in \mathbb{R}^{d_h}$ ) learned from the previous  $t-1$  features, i.e.  $(\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior}) = \phi_{prior}(\mathbf{h}_{t-1})$ . Note that  $\phi_{prior} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$  refers to two separate MLPs, one to obtain  $\boldsymbol{\mu}_{t,prior}$  and another with *softplus* activation to estimate the standard deviation ( $\boldsymbol{\sigma}_{t,prior}$ ). Unless otherwise specified, all MLPs are two layered neural networks with ReLU activation.

The posterior distribution of the abstract goal  $r(\mathbf{z}_t | \mathbf{x}_{1:t}) \sim \mathcal{N}(\boldsymbol{\mu}_{t,pos}, \boldsymbol{\sigma}_{t,pos})$  computes the effect of observing the incoming new feature  $\mathbf{x}_t$ . The parameters

of posterior distribution  $r$  are computed as follows:

$$(\boldsymbol{\mu}_{t,pos}, \boldsymbol{\sigma}_{t,pos}) = \phi_{pos}([\phi_x(\mathbf{x}_t), \phi_h(\mathbf{h}_{t-1})]), \quad (1)$$

where  $\phi_{pos} : \mathbb{R}^{2 \times d_z} \rightarrow \mathbb{R}^{d_z}$ ,  $\phi_x : \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_z}$ ,  $\phi_h : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$  are linear layers and  $[\cdot, \cdot]$  represents vector concatenation. We use the reparameterization trick [17] to sample an abstract goal ( $\mathbf{z}_t \in \mathbb{R}^{d_z}$ ) from the prior distribution  $q(\mathbf{z}_t | \mathbf{x}_{1:t-1})$  as follows:

$$\mathbf{z}_t = \boldsymbol{\mu}_{t,prior} + \boldsymbol{\sigma}_{t,prior} \odot \boldsymbol{\epsilon}, \quad (2)$$

where  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^{d_z}$  is a standard Gaussian distribution. Then sampled  $\mathbf{z}_t$  is used to obtain the next hidden state of the RNN<sup>1</sup> as follows:

$$\mathbf{h}_t = RNN(\mathbf{h}_{t-1}, [\phi_x(\mathbf{x}_t), \phi_z(\mathbf{z}_t)]), \forall t \in 1, \dots, T \quad (3)$$

where  $\phi_z : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_z}$  acts as a feature extractor over  $\mathbf{z}_t$ . The sampled abstract goal ( $\mathbf{z}_t$ ) can be used to reconstruct (or generate) the feature sequence as done in VRNN framework [4, 9]. However, we use it to represent feature-based abstract goal. Our intuition comes from the fact that humans derive action plans from goals, and videos are a realization of this action plan. Therefore, by construction, goal determines the video (feature evolution in our case). Interestingly, as the abstract goal latent variable encapsulates the video feature generation process, by analogical similarity, we make the proposition that latent variable ( $\mathbf{z}_t$ ) represents the notion of feature-based abstract goal.

Therefore, we denote the “feature-based abstract goal distribution” as follows:

$$p(\mathbf{z}_T) = q(\mathbf{z}_T | \mathbf{x}_{1:T-1}). \quad (4)$$

The abstract goal distribution *represents all abstract goals with respect to a particular observed feature sequence*. Any observed action may lead to more than one goal. Our abstract goal representation captures these variations.

### 3.2 Action representations

Human actions are causal in nature and the next action in a sequence depends on the earlier actions. For example, *washing vegetables* is succeeded by *cutting vegetables* when the goal is “making a salad”. We capture the causality between observed and next actions using “the observed action representation” and the “next action representation”. We obtain the **observed action representation** ( $\mathbf{a}_O$ ) using feature-based abstract goal and the hidden state of RNN as follows:

$$\mathbf{a}_O = \phi_O([\phi_z(\mathbf{z}_T), \phi_h(\mathbf{h}_T)]). \quad (5)$$

Here  $\phi_O : \mathbb{R}^{2 \times d_z} \rightarrow \mathbb{R}^{d_h}$  and  $\mathbf{z}_T$  is sampled from the abstract goal distribution  $p(\mathbf{z}_T)$  using Equation 2.

<sup>1</sup> Our RNN is a standard GRU cell.

Then we obtain the distribution of **next action representation** ( $\mathbf{a}_N$ ) conditioned on the hidden state of the RNN and the observed action representation denoted by  $p(\mathbf{a}_N|\mathbf{h}_T, \mathbf{a}_O)$ . The reason for modeling next action representation as a distribution conditioned on hidden state and the observed action representation is two-fold. First, a particular observed action may lead to different next actions depending on the context and goal. Note that in our model, both observed action representation  $\mathbf{a}_O$  and the RNN hidden state  $\mathbf{h}_T$  depend on the feature-based abstract goal representation. Second, there can be variations in human behavior when executing the same task. The next action representations are generated using a Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N}^2)$  where  $\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N} \in \mathbb{R}^{d_z}$ . The parameters of next action distribution are estimated as

$$p(\mathbf{a}_N|\mathbf{h}_T, \mathbf{a}_O) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N}^2), \quad (6)$$

where  $(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N}) = \phi_N([\phi_h(\mathbf{h}_T), \phi_a(\mathbf{a}_O)])$ . The mapping network  $\phi_a : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$  and  $\phi_N : \mathbb{R}^{2 \times d_z} \rightarrow \mathbb{R}^{d_z}$  are two separate MLPs. Now we sample multiple next action representations from the next action representation distribution using the reparameterization trick as in Equation 7,

$$\mathbf{a}_N = \boldsymbol{\mu}_{\mathbf{a}_N} + \boldsymbol{\sigma}_{\mathbf{a}_N} \odot \boldsymbol{\epsilon}, \quad (7)$$

where  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^{d_z}$  is a standard Gaussian distribution.

### 3.3 Action-based abstract goal representation

Now, we obtain *action-based abstract goal* from observed and next action representations using generative variational framework [17]. The distribution for action-based abstract goal is modeled with a Gaussian distribution conditioned on the next action representation denoted by  $q(\mathbf{z}_N|\mathbf{a}_N)$  whose parameters are computed as  $q(\mathbf{z}_N|\mathbf{a}_N) \sim \mathcal{N}(\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq})$  where  $(\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq}) = \phi_{Nq}(\phi_a(\mathbf{a}_N))$  and  $\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq} \in \mathbb{R}^{d_z}$  and  $\phi_{Nq} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$  is implemented with two MLPs. On the other hand, parameters of the action-based abstract goal distribution ( $r$ ) conditioned on both observed and next action representation are given as  $r(\mathbf{z}_N|\mathbf{a}_N, \mathbf{a}_O) \sim \mathcal{N}(\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr})$  whose parameters are estimated as:

$$(\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr}) = \phi_{Nr}([\phi_a(\mathbf{a}_N), \phi_a(\mathbf{a}_O)]) \quad (8)$$

where  $\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr} \in \mathbb{R}^{d_z}$  and  $\phi_{Nr} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$  is a dual headed MLP. Finally, the **action-based abstract goal** distribution for the next action  $p(\mathbf{z}_N)$  is given by the distribution

$$p(\mathbf{z}_N) = q(\mathbf{z}_N|\mathbf{a}_N). \quad (9)$$

We use both feature-based and action-based abstract goal representation to find the *best candidate for next action* as explained in next section. It should be noted that while the  $q(\mathbf{z}_N|\mathbf{a}_N)$  only depends on next action representation and  $r(\mathbf{z}_N|\mathbf{a}_N, \mathbf{a}_O)$  depends on both observed and next action representation. As  $r()$  has more evidence compared to  $q()$ ,  $r()$  acts as the posterior distribution in our modeling.

### 3.4 Next action anticipation with goal consistency

Given a sampled feature-based abstract goal  $\mathbf{z}_T$ , we select the best next action representation  $\mathbf{a}_N^*$  using the divergence between  $p(\mathbf{z}_T)$  distribution (eq. 4) and  $p(\mathbf{z}_N)$  distribution (eq. 9). We call this divergence as the **goal consistency criterion**. For a given  $\mathbf{z}_T$ , observed action  $\mathbf{a}_O$  and the next sampled action  $\mathbf{a}_N$ , the goal consistency criterion is derived from the average of KL-divergence  $D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N))$  and  $D_{KL}(p(\mathbf{z}_N)||p(\mathbf{z}_T))$  as follows:

$$D(\mathbf{a}_N) = \frac{D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N)) + D_{KL}(p(\mathbf{z}_N)||p(\mathbf{z}_T))}{2}. \quad (10)$$

We choose the best next action candidate (i.e. the anticipated action candidate representation)  $\mathbf{a}_N^*$  that minimizes the goal consistency criterion. The rationale is that the best anticipated action should have an action-based abstract goal representation  $p(\mathbf{z}_N)$  that aligns with the feature-based abstract goal distribution  $p(\mathbf{z}_T)$ . We use the following algorithm to find the best next action candidate  $\mathbf{a}_N^*$ .

---

#### Algorithm 1 Best next action selection

---

- 1: Sample feat-based abstract goal  $\mathbf{z}_t$  from eq. 4  $\rightarrow \mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{x}_{1:t-1})$
  - 2: Get observed action representation  $\mathbf{a}_O$  (eq. 5)
  - 3: Get next action representation distribution  $p(\mathbf{a}_N|\mathbf{h}_t, \mathbf{a}_O)$  (eq. 6)
  - 4: Sample  $K$  next action representations  $\mathcal{N} = \{\mathbf{a}_N^1, \dots, \mathbf{a}_N^K\} \sim p(\mathbf{a}_N|\mathbf{h}_t, \mathbf{a}_O)$
  - 5: Best next action  $\mathbf{a}_N^* = \arg \min_{\mathbf{a}_N^k \in \mathcal{N}} D(\mathbf{a}_N^k); k \in \{1, \dots, K\}$
- 

Finally, we predict the anticipated action from the selected next action representation as  $\hat{\mathbf{y}} = \phi_c(\mathbf{a}_N^*)$ . where  $\phi_c : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_c}$  is the MLP classifier and  $\hat{\mathbf{y}}$  is the class score vector. It should be noted that in Algorithm 1, we sample only one feature-based abstraction goal in line 1 of the algorithm. However, during training we sample  $Q$  number of feature-based abstraction goals and for each of them we sample  $K$  number of next action representations. In this case, we select the best candidate from all  $K \times Q$  next action representation candidates using Equation 10. Therefore, the next best action is consistent and does not rely too much on sampling as long as we sample sufficient candidate next actions.

Even if the feature-based abstract goal  $P(\mathbf{z}_T)$  is obtained from VRNN framework [4, 9], the formulation of action representations  $\mathbf{a}_O$  and  $\mathbf{a}_N$ , action-based abstract goal  $P(\mathbf{z}_N)$  and goal consistency criterion is drastically different from [1, 22]. In [27], goal consistency is defined between latent goals before and after the action using a hard threshold. Instead, our goal consistency is a symmetric KL divergence between  $p(\mathbf{z}_T)$  and  $p(\mathbf{z}_N)$  distributions which aims to align the two abstract goal distributions. This also results in a massive improvement in next action anticipation performance as shown in the experiments.

### 3.5 Loss functions and training of our model

Our anticipation network is trained using a number of losses. In contrast to prior stochastic methods [1, 21, 22], we introduce three KL divergence losses, based on a) feature-based abstract goal ( $\mathcal{L}_{OG}$ ), b) action-based abstract goal ( $\mathcal{L}_{NG}$ ), and c) goal-consistency ( $\mathcal{L}_{GC}$ ). The first loss function is used to learn the parameters of the feature-based abstract goal distribution. We compute the KL-divergence between the conditional prior  $q(\mathbf{z}_t|\mathbf{x}_{1:t-1})$  and posterior  $r(\mathbf{z}_t|\mathbf{x}_{1:t})$  distributions for every feature in the observed feature sequence and minimize the sum given as follows  $\mathcal{L}_{OG} = \sum_{t=1}^T D_{KL}(r(\mathbf{z}_t|\mathbf{x}_{1:t})||q(\mathbf{z}_t|\mathbf{x}_{1:t-1}))$  and we call this **observed goal loss**. This loss is based on the intuition that the abstract goal should not change due to a new observed feature.

Our second loss arises when we learn the action-based abstract goal distribution. We compute the KL-divergence between  $r(\mathbf{z}_N|\mathbf{a}_N^*, \mathbf{a}_O)$  and  $q(\mathbf{z}_N|\mathbf{a}_N^*)$  distributions of action-based abstract goal distributions as  $\mathcal{L}_{NG} = D_{KL}(r(\mathbf{z}_N|\mathbf{a}_N^*, \mathbf{a}_O)||q(\mathbf{z}_N|\mathbf{a}_N^*))$ . We denote the corresponding best action-based abstract goal distribution by  $p(\mathbf{z}_N^*) = q(\mathbf{z}_N|\mathbf{a}_N^*)$ . The intuition is same as before, the goal should not change because of the next best action  $\mathbf{a}_N^*$ .

Furthermore, the feature-based and action-based abstract goal distributions should be aligned with respect to the selected next best action  $\mathbf{a}_N^*$ . Therefore, we minimize the symmetric KL-Divergence between the feature-based and best-action-based abstract goal distribution as follows:

$$\mathcal{L}_{GC} = \frac{D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N^*)) + D_{KL}(p(\mathbf{z}_N^*)||p(\mathbf{z}_T))}{2}. \quad (11)$$

We coin this loss as **goal consistency loss**. This loss is based on  $D(\mathbf{a}_N)$  in Equation 10 with the only difference being that  $p(\mathbf{z}_N^*) = q(\mathbf{z}_N|\mathbf{a}_N^*)$  is computed with respect to the selected best next action representation  $\mathbf{a}_N^*$ . Finally, we have the cross-entropy loss for comparing the model’s prediction  $\hat{\mathbf{y}}$  with the ground truth one-hot label  $\mathbf{y}$  as  $\mathcal{L}_{NA} = -\sum \mathbf{y} \odot \log(\hat{\mathbf{y}})$ . The loss function to train the model is a combination of all losses given as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{OG} + \mathcal{L}_{NG} + \mathcal{L}_{GC} + \mathcal{L}_{NA}. \quad (12)$$

We experimented with adding different weights to each loss but there is no significant difference in performance. Therefore, we weigh them equally.

## 4 Experiments and results

### 4.1 Datasets, features, and training details

We use well known action anticipation datasets, *Epic-Kitchens55*[5] (EK55) and *Epic-Kitchens100*[6] (EK100) to evaluate our approach.

We validate our models using the TSN features obtained from RGB and optical flow videos, and bag of object features provided by [10] for a fair comparison with existing approaches. Our base model has the following parameters: observed

**Table 1.** Comparison of anticipation accuracy with state-of-the-art on EK55 evaluation server with anticipation time of 1 sec. ACT: for action.

Method	Top-1 accuracy(%)			Top-5 accuracy(%)			Precision(%)			Recall(%)		
	VERB	NOUN	ACT.	VERB	NOUN	ACT.	VERB	NOUN	ACT	VERB	NOUN	ACT.
<b>Seen Kitchens (S1)</b>												
RU-LSTM [10]	33.04	22.78	14.39	79.55	50.95	33.73	25.50	24.12	07.37	15.73	19.81	07.66
Lat. Goal [27]	27.96	27.40	08.10	78.09	55.98	26.46	-	-	-	-	-	-
SRL [25]	34.89	22.84	14.24	79.59	52.03	34.61	28.29	25.69	06.45	12.19	19.16	06.34
ImagineRNN [33]	35.44	22.79	14.66	79.72	52.09	34.98	28.04	24.18	06.66	16.03	19.61	07.08
Temp. Agg. [29]	37.87	24.10	16.64	79.74	53.98	36.06	<b>36.41</b>	25.20	09.64	15.67	22.01	10.05
MM-Trans [26]	28.59	27.18	10.85	78.64	57.66	30.83	17.50	26.20	03.81	10.81	24.89	04.49
MM-TCN [35]	37.16	23.75	15.45	79.48	51.86	34.37	28.18	23.82	06.94	16.05	22.31	08.40
AVT [13]	34.36	20.16	16.84	80.03	51.57	36.52	23.25	17.77	09.71	14.02	18.81	10.11
DCR [34]	-	-	17.70	-	-	<b>38.50</b>	-	-	-	-	-	-
<b>Abstract Goal (VRNN)</b>	<b>51.56</b>	<b>35.34</b>	<b>22.03</b>	<b>82.56</b>	<b>58.01</b>	38.29	34.83	<b>31.33</b>	<b>13.08</b>	<b>26.67</b>	<b>31.42</b>	<b>12.20</b>
<b>Unseen Kitchens (S2)</b>												
RU-LSTM [10]	27.01	15.19	08.16	69.55	34.38	21.10	13.69	09.87	03.64	09.21	11.97	04.83
Lat. Goal [27]	22.40	19.12	04.78	72.07	42.68	16.97	-	-	-	-	-	-
SRL [25]	27.42	15.47	08.88	71.90	36.80	22.06	20.23	12.48	02.84	07.83	12.25	04.33
ImagineRNN [33]	29.33	15.50	09.25	70.67	35.78	22.19	17.10	12.20	03.47	09.66	12.36	05.21
Temp. Agg. [29]	29.50	16.52	10.04	70.13	37.83	23.42	20.43	12.95	04.92	08.03	12.84	06.26
MM-Trans [26]	26.80	18.40	06.76	70.40	<b>44.18</b>	20.04	09.53	15.17	02.23	07.73	15.19	03.34
MM-TCN [35]	30.66	14.92	08.91	72.00	36.67	21.68	10.51	12.26	04.35	09.79	12.72	04.94
AVT [13]	30.66	15.64	10.41	72.17	40.76	24.27	12.86	11.83	04.84	09.89	13.46	06.41
DCR [34]	-	-	10.90	-	-	<b>24.80</b>	-	-	-	-	-	-
<b>Abstract Goal (VRNN)</b>	<b>41.41</b>	<b>22.36</b>	<b>13.28</b>	<b>73.10</b>	41.62	24.24	<b>23.62</b>	<b>18.29</b>	<b>08.73</b>	<b>15.70</b>	<b>18.29</b>	<b>08.29</b>

duration - 2 seconds, frame rate - 3 fps, RNN (GRU) hidden dimension  $d_h = 256$ , abstract goal dimension  $d_z = 128$ , number of sampled feature-based abstract goals ( $Q = 3$ ), number of next-action-representation candidates ( $K = 10$ ),  $\mathcal{L}_{total}$  loss, and fixed anticipation time - 1s (following EK55 and EK100 evaluation server criteria), unless specified otherwise. We use a batch size of 128 videos and train for 15 epochs with a learning rate of 0.001 using Adam optimizer with weight decay (AdamW) in Pytorch. All our MLPs have 256 hidden dimensions.

## 4.2 Comparison with state-of-the-art

We compare the performance of Abstract Goal (our method) with current state-of-the-art approaches on both the seen and unseen test sets of EK55 datasets in Table 1 using a late fusion of TSN-RGB, TSN-Flow, and Object features like most of the prior work. We train separate models for verb and noun anticipation and combine their predictions to obtain action anticipation accuracy. The model structure is the same for both the verb and noun models but the final classification output is either verb or noun. Our method outperforms all other prior state-of-the-art methods for both seen kitchens (S1) and unseen kitchens (S2). Notably, we outperform Transformer-based AVT [13] and Temporal-Aggregation [29] in all measures in both seen and unseen kitchens except for Top-5 accuracy on unseen kitchens. We believe this improvement is due to two factors, (i) stochastic modeling is massively important for action anticipation, and (ii) the effective use of goal information is paramount for better action anticipation.

**Table 2.** Comparison on EK100 dataset on evaluation server using test set. Accuracy measured by mean recall@5 (%) following the standard protocol.

Method	Input	Overall			Unseen Kitchens			Tail Classes		
		VERB	NOUN	ACT.	VERB	NOUN	ACT.	VERB	NOUN	ACT.
AVT [13]	Frames	26.69	32.33	16.74	21.03	27.64	12.89	19.28	24.03	13.81
RAFTformer [12]	Frames	30.10	34.10	15.40	-	-	-	-	-	-
InAViT [28]	Frames	<b>49.14</b>	<b>49.97</b>	<b>23.75</b>	<b>44.36</b>	<b>49.28</b>	<b>23.49</b>	<b>43.17</b>	<b>39.91</b>	<b>18.11</b>
RU-LSTM [6]	TSN	25.25	26.69	11.19	19.36	26.87	09.65	17.56	15.97	07.92
Temp. Agg. [29]	TSN	21.76	30.59	12.55	17.86	27.04	10.46	13.59	20.62	08.85
TransAction [15]	TSN	36.15	32.20	13.39	27.60	24.24	10.05	<b>32.06</b>	<b>29.87</b>	11.88
DCR[34]	TSN	-	-	17.30	-	-	14.10	-	-	<b>14.30</b>
<b>Abstract Goal (VRNN)</b>	TSN	31.40	30.10	14.29	31.36	35.56	<b>17.34</b>	22.90	16.42	07.70
<b>Abstract Goal (TF)</b>	TSN	<b>37.63</b>	<b>38.70</b>	14.21	<b>34.92</b>	<b>38.88</b>	14.25	30.67	29.10	09.11

Despite, these excellent results on EK55, our overall results on EK100 are not state-of-the-art—see Table 2. Our method performs not as well as recent methods that are extensively pre-trained vision transformer (ViT) models with image and action recognition datasets before being trained for action anticipation [12, 13, 28]. On the other hand, our model is trained directly on the target dataset using temporal segment network (TSN) [31] features. Compared to the best Transformer model [15, 34] trained on TSN features, Abstract Goal - VRNN performs better on both overall and unseen kitchens of the EK100 dataset but not as well on tail classes. EK100 dataset is dominated by long-tailed distribution where 228 noun classes out of 300 are in the tail classes. Similarly, 86 verbs out of 97 are in the tail classes. In our model, the next-action-representation is modeled with a Gaussian distribution (Equation 6), and therefore, it is not able to cater to exceptionally long tail class distributions as in EK100. This is a limitation of our method. We do not witness the tail-class issue in EK55 as the performance measure used is accuracy compared to mean-recall in EK100. Accuracy is influenced heavily by frequent classes but mean-recall treats all classes equally.

For completeness, we test whether the tail class issue on EK100 can be resolved using a transformer network (6 layers with 8 attention heads) instead of a GRU for observed feature summarization. While abstract goal with transformer (TF) improves tail class performance it is not able to outperform [15, 34] on tail classes. This confirms our hypothesis that using Gaussian distribution for next-action-representation (action-based abstract goal) can limit tail class performance but improves overall and unseen kitchens anticipation accuracy.

### 4.3 Impact of goal consistency criterion and loss

In this section, we evaluate the impact of Goal Consistency (GC) criterion and the loss derived from it  $\mathcal{L}_{gc}$  using the validation set of EK55 and EK100 datasets. We train separate models for verb and noun anticipation using TSN-RGB (RGB) and Object (OBJ) features, respectively. As Mean and Median sampling are used



in prior variational prediction models [1], here we use mean and median sampling as two baselines to show the effect of GC. We sample  $Q \times K$  number of next-action representations ( $\mathbf{a}_N$ ) instead of selecting the best next-action candidate using GC (Algorithm 1). Then we obtain the mean/median vector of all sampled candidates and then make the prediction using the classifier (e.g. mean vector =  $\frac{\sum \mathbf{a}_N}{Q \times K}$ ). We also experimented with a majority/median class prediction baseline. In this case, we take all  $Q \times K$  predictions from the classifier (from the next action-representation candidates) and pick the majority/median class as the final prediction. Everything else stays the same for all these mean/majority/median baseline models, except we do not use the GC criterion (Equation 10) and the goal consistency loss  $\mathcal{L}_{gc}$ . Results are reported in Table 3.

**Table 3.** The impact of goal consistency criterion and loss. @1 and @5 denotes Top-1 and Top-5 accuracy and V stands for verb and N stands for noun.

Goal candidate (Q) & Action candidate (K)		EK55				EK100			
		V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
Mean	Q=1, K=10	41.79	72.23	25.79	49.50	44.51	76.89	22.72	50.78
Median		41.16	71.32	24.30	48.31	45.44	77.91	22.15	51.23
Majority class		41.98	72.89	25.98	50.01	42.98	74.56	24.13	53.45
Median class		41.02	72.11	22.88	49.87	44.19	77.00	22.97	51.98
Our model		<b>45.18</b>	<b>77.30</b>	<b>28.16</b>	<b>51.08</b>	<b>48.84</b>	<b>80.52</b>	<b>27.50</b>	<b>55.83</b>
Mean	Q=3, K=10	39.40	72.23	24.22	48.96	45.90	77.88	22.41	50.87
Median		41.32	71.32	26.60	51.70	45.63	77.02	24.33	52.87
Majority class		38.39	69.42	24.70	48.22	45.72	78.61	22.61	50.89
Median class		40.43	71.43	26.52	52.33	45.84	78.09	23.78	52.33
Our model		<b>44.68</b>	<b>77.14</b>	<b>28.29</b>	<b>53.78</b>	<b>49.02</b>	<b>80.86</b>	<b>28.52</b>	<b>54.91</b>
Without $\mathcal{L}_{GC}$	Q=1, K=1	38.31	70.77	19.74	43.11	43.82	77.45	21.25	51.99
With $\mathcal{L}_{GC}$		<b>40.88</b>	<b>71.43</b>	<b>22.09</b>	<b>46.29</b>	<b>46.80</b>	<b>78.41</b>	<b>26.80</b>	<b>53.32</b>

As can be seen from the results, there is a significant impact of GC. Especially, there is an improvement of 3.39% and 2.37% for top-1 verb and noun accuracy respectively using our GC model in the EK55 dataset for  $Q = 1, K = 10$  over Mean sampling baseline. A similar trend can be seen for EK100 and  $Q = 3, K = 10$  as well. Our model also outperforms majority and median class sampling baselines for both  $[Q = 1, K = 10]$  and  $[Q = 3, K = 10]$  configurations indicating the effectiveness of goal consistency both as GC criterion and GC loss  $\mathcal{L}_{GC}$ . Overall, our method with GC loss and criterion performs better than all other variants. Perhaps this is because the GC criterion allows the model to regularize the candidate selection while GC loss allows the model to enforce this during the training. This clearly shows the impact of *goal consistency formulation* of our model for action anticipation.

We perform a more controlled experiment to further evaluate the impact of GC loss where we set  $Q = 1$  and  $K = 1$  and train our model with and without

**Table 4.** Ablation on the sensitivity of number of sampled feature-based-abstract-goals ( $Q$ ) and next-action representation candidate  $K$  on EK55 and EK100 validation set.

parameter	value	EK55				EK100			
		V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
<b>num. feature-based abstract goals (Q)</b> ( $K = 10$ )	1	45.18	77.30	28.16	51.08	48.84	80.52	27.50	<b>55.83</b>
	2	44.44	76.19	<b>28.47</b>	52.38	49.25	80.44	28.41	55.65
	3	44.68	77.14	28.29	<b>53.78</b>	49.02	<b>80.86</b>	<b>28.52</b>	54.91
	4	45.31	<b>77.91</b>	26.28	50.33	48.86	80.46	28.16	55.11
	5	<b>45.80</b>	77.40	26.95	51.93	<b>49.71</b>	80.40	28.04	55.16
	10	<b>44.68</b>	77.14	<b>28.29</b>	<b>53.78</b>	49.02	80.86	<b>28.52</b>	<b>54.91</b>
<b>num. next action candidates (K)</b> ( $Q=3$ )	1	39.81	72.31	21.48	44.96	44.24	75.67	20.06	42.56
	3	40.49	74.20	22.60	46.22	44.37	76.11	21.07	44.51
	5	41.32	74.26	23.17	48.23	45.61	78.91	22.91	45.12
	10	<b>44.68</b>	77.14	<b>28.29</b>	<b>53.78</b>	49.02	80.86	<b>28.52</b>	<b>54.91</b>
	20	43.79	<b>79.00</b>	27.07	51.10	49.01	80.36	28.13	55.40
	30	44.56	77.81	27.80	51.00	<b>49.18</b>	<b>81.20</b>	27.44	53.42

GC loss ( $\mathcal{L}_{GC}$ ). It should be noted that when  $Q = 1$  and  $K = 1$ , GC criterion has no impact because we do not have multiple candidates to evaluate. The only meaningful way to see the effect of GC is to compare a model trained with and without the GC loss. To obtain a statistically meaningful result, we repeat this experiment 10 times and report the mean performance. As it can be seen from the results in Table 3 (last two rows), clearly GC loss has a positive impact even when we just sample a single action candidate from our stochastic model. We see that compared to our model variant [ $Q = 1, K = 1$  with  $\mathcal{L}_{GC}$ ], the [ $Q = 1, K = 10$  with  $\mathcal{L}_{GC}$ ] model performs significantly better (last row vs row 5 of Table 3). This indicates the impact of next-action-representation sampling (Equation 6) even for a single sampled feature-based abstract goal ( $Q = 1$ ). We conclude that the goal consistency loss, the goal consistency criterion, and next-action-representation distribution modeling (all novel concepts introduced in this paper) are effective for action anticipation.

**Table 5.** Loss ablation on EK55 and EK100 validation set. *i.e.*  $\mathcal{L}_{NA}$ -Next action cross-entropy loss,  $\mathcal{L}_{OG}$ -Feature-based abstract goal loss,  $\mathcal{L}_{NG}$ -Action-based abstract goal loss,  $\mathcal{L}_{GC}$ -Goal consistency loss.

Losses	EK55				EK100			
	V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
$\mathcal{L}_{NA}$	21.36	69.69	27.76	51.89	24.46	72.31	27.12	54.55
$\mathcal{L}_{NA} + \mathcal{L}_{OG}$	44.42	77.79	28.41	51.31	43.23	75.63	23.45	52.89
$\mathcal{L}_{NA} + \mathcal{L}_{NG}$	46.01	77.94	29.05	52.32	46.94	78.44	22.96	49.66
$\mathcal{L}_{NA} + \mathcal{L}_{GC}$	43.83	77.43	28.06	51.87	44.45	76.72	20.31	47.87
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{NG}$	44.47	77.12	28.51	51.34	46.73	78.62	24.56	51.33
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{GC}$	45.47	77.42	28.61	52.34	47.25	78.11	26.91	53.34
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{NG} + \mathcal{L}_{GC}$	<b>46.37</b>	<b>77.97</b>	<b>29.86</b>	<b>52.74</b>	<b>49.02</b>	<b>80.86</b>	<b>28.52</b>	<b>54.91</b>

Apart from GC loss, we also study the impact of other loss functions described in Section 3.5 and report the results in Table 5. If we use only the supervised

cross-entropy loss (i.e.,  $\mathcal{L}_{NA}$ ), then the performance is the worst, especially for verbs. Both  $\mathcal{L}_{OG}$  and  $\mathcal{L}_{NG}$  help in regularizing the abstract goal representations ( $\mathbf{z}_t$  and  $\mathbf{a}_N$ ), and therefore results improve significantly. Especially, the  $\mathcal{L}_{NA} + \mathcal{L}_{NG}$  is the best loss combination for a pair of losses. When we combine all four losses, we get the best results. While  $\mathcal{L}_{NA} + \mathcal{L}_{NG}$  regularizes the learning of abstract goal representations,  $\mathcal{L}_{GC}$  which minimizes the divergence between feature-based and action-based goal distributions improves the choice of next verb or noun among the plausible candidates. We conclude that all four losses are important for our model.

#### 4.4 Effect of action-based abstract goal distributions

We demonstrate the efficacy of action-based abstract goal in our model by comparing it to a variant of our model having only the feature-based abstract goal (equivalent to a VRNN) in Table 6. For the feature-based abstract goal (Feat. abs. goal), we obtain a latent variable  $\mathbf{z}_T$  and the observed action representation  $\mathbf{a}_O$  from Equation 5. We classify  $\mathbf{a}_O$  using a classifier to obtain the future action and train using cross-entropy loss and KL-divergence ( $\mathcal{L}_{OG}$ ). We do not have

**Table 6.** Effect of action-based abstract goal

Model	V@1	V@5	N@1	N@5
Abs. Goal (Feat)-mean	27.76	61.23	22.34	46.78
Abs. Goal (Feat)-median	38.13	68.94	23.85	47.56
Abs. Goal (Feat+Act)-mean	39.40	72.23	24.22	48.96
Abs. goal (Feat+Act)-median	<b>44.68</b>	<b>77.14</b>	<b>28.29</b>	<b>53.78</b>

GC criterion when using only the feature abstract goal distribution and hence we sample 30 candidates for  $\mathbf{a}_O$  and consider their mean or median. The number of sampled candidates is chosen to match our feature + action abstract goal model with 30 next action candidates ( $Q = 3, K = 10$ ). As shown in Table 6, using action-based abstract goal in conjunction with feature-based abstract goal performs much better than only feature abstract goal distribution (under both mean or median prediction).

## 5 Conclusion

We present a novel approach for action anticipation where abstract goals are learned with a stochastic recurrent model. We outperform existing approaches on EK55 and our model generalizes to unseen kitchen environments in both EK55 and EK100 datasets. We also show the importance of goal consistency criterion, goal consistency loss, next-action representation modeling, and architecture. One limitation of the current work is the inability to directly interpret the latent goal representation learned by our model. Second, our method is not able to tackle long-tail-class distribution issues. In the future, we aim to address these limitations of our model.

**Acknowledgment.** This research/project is supported by the National Research Foundation, Singapore, under its NRF Fellowship (Award# NRF-NRFF14-2022-0001) and by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). This research is also supported by funding allocation to B.F. by the Agency for Science, Technology and Research (A\*STAR) under its SERC Central Research Fund (CRF), as well as its Centre for Frontier AI Research (CFAR).

## References

1. Abu Farha, Y., Gall, J.: Uncertainty-aware anticipation of activities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
2. Abu Farha, Y., Richard, A., Gall, J.: When will you do what?-anticipating temporal occurrences of activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5343–5352 (2018)
3. Chang, C.Y., Huang, D.A., Xu, D., Adeli, E., Fei-Fei, L., Niebles, J.C.: Procedure planning in instructional videos. In: European Conference on Computer Vision. pp. 334–350. Springer (2020)
4. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. *Adv. Neural. Inf. Process. Syst.* **28**, 2980–2988 (2015)
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020)
6. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision* pp. 1–23 (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
8. Fernando, B., Herath, S.: Anticipating human actions by correlating past with the future with jaccard similarity measures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13224–13233 (2021)
9. Fraccaro, M., Sønderby, S.K., Paquet, U., Winther, O.: Sequential neural models with stochastic layers. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 2207–2215 (2016)
10. Furnari, A., Farinella, G.: Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020)
11. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Forecasting future action sequences with neural memory networks. In: 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019. p. 298. BMVA Press (2019), <https://bmvc2019.org/wp-content/uploads/papers/0585-paper.pdf>
12. Girase, H., Agarwal, N., Choi, C., Mangalam, K.: Latency matters: Real-time action forecasting transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18759–18769 (2023)

13. Girdhar, R., Grauman, K.: Anticipative Video Transformer. In: ICCV (2021)
14. Gong, D., Lee, J., Kim, M., Ha, S.J., Cho, M.: Future transformer for long-term action anticipation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3052–3061 (2022)
15. Gu, X., Qiu, J., Guo, Y., Lo, B., Yang, G.: Transaction: ICL-SJTU submission to epic-kitchens action anticipation challenge 2021. CoRR **abs/2107.13259** (2021)
16. Ke, Q., Fritz, M., Schiele, B.: Time-conditioned action anticipation in one shot. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9925–9934 (2019)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
18. Liu, M., Tang, S., Li, Y., Rehg, J.M.: Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In: European Conference on Computer Vision. pp. 704–721. Springer (2020)
19. Liu, T., Lam, K.M.: A hybrid egocentric activity anticipation framework via memory-augmented recurrent and one-shot representation forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13904–13913 (2022)
20. Loh, S.B., Roy, D., Fernando, B.: Long-term action forecasting using multi-headed attention-based variational recurrent neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2419–2427 (2022)
21. Mascaró, E.V., Ahn, H., Lee, D.: Intention-conditioned long-term human egocentric action anticipation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6048–6057 (2023)
22. Mehrasa, N., Jyothi, A.A., Durand, T., He, J., Sigal, L., Mori, G.: A variational auto-encoder model for stochastic point processes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3165–3174 (2019)
23. Miech, A., Laptev, I., Sivic, J., Wang, H., Torresani, L., Tran, D.: Leveraging the present to anticipate the future in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
24. Nawhal, M., Jyothi, A.A., Mori, G.: Rethinking learning approaches for long-term action anticipation. In: European Conference on Computer Vision. pp. 558–576. Springer (2022)
25. Qi, Z., Wang, S., Su, C., Su, L., Huang, Q., Tian, Q.: Self-regulated learning for egocentric video activity anticipation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
26. Roy, D., Fernando, B.: Action anticipation using pairwise human-object interactions and transformers. IEEE Transactions on Image Processing (2021)
27. Roy, D., Fernando, B.: Action anticipation using latent goal learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2745–2753 (January 2022)
28. Roy, D., Rajendiran, R., Fernando, B.: Interaction region visual transformer for egocentric action anticipation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6740–6750 (2024)
29. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: European Conference on Computer Vision. pp. 154–171. Springer (2020)

30. Song, Y., Byrne, E., Nagarajan, T., Wang, H., Martin, M., Torresani, L.: Ego4d goal-step: Toward hierarchical understanding of procedural activities. *Advances in Neural Information Processing Systems* **36** (2024)
31. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: *European conference on computer vision*. pp. 20–36. Springer (2016)
32. Wu, C.Y., Li, Y., Mangalam, K., Fan, H., Xiong, B., Malik, J., Feichtenhofer, C.: Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. *arXiv preprint [arXiv:2201.08383](https://arxiv.org/abs/2201.08383)* (2022)
33. Wu, Y., Zhu, L., Wang, X., Yang, Y., Wu, F.: Learning to anticipate egocentric actions by imagination. *IEEE Trans. Image Process.* **30**, 1143–1152 (2021). <https://doi.org/10.1109/TIP.2020.3040521>
34. Xu, X., Li, Y.L., Lu, C.: Learning to anticipate future with dynamic context removal. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 12734–12744 (2022)
35. Zatsarynna, O., Abu Farha, Y., Gall, J.: Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2249–2258 (2021)
36. Zhang, T., Min, W., Yang, J., Liu, T., Jiang, S., Rui, Y.: What if we could not see? counterfactual analysis for egocentric action anticipation. In: *IJCAI* (2021)
37. Zhao, Q., Wang, S., Zhang, C., Fu, C., Do, M.Q., Agarwal, N., Lee, K., Sun, C.: Antgpt: Can large language models help long-term action anticipation from videos? In: *The Twelfth International Conference on Learning Representations* (2023)



# SHARP: Segmentation of Hands and Arms by Range Using Pseudo-depth for Enhanced Egocentric 3D Hand Pose Estimation and Action Recognition

Wiktoria Mucha<sup>1</sup>, Michael Wray<sup>2</sup>, and Martin Kampel<sup>1</sup>

<sup>1</sup> Computer Vision Lab, TU Wien, Favoritenstr. 9/193-1, 1040 Vienna, Austria

{wiktoria.mucha,martin.kampel}@tuwien.ac.at

<sup>2</sup> University of Bristol, Bristol, UK

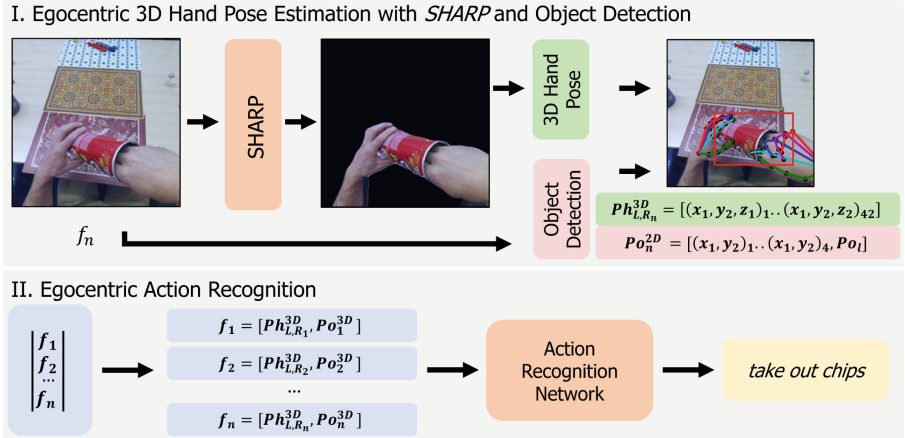
michael.wray@bristol.ac.uk

**Abstract.** Hand pose represents key information for action recognition in the egocentric perspective, where the user is interacting with objects. We propose to improve egocentric 3D hand pose estimation based on RGB frames only by using pseudo-depth images. Incorporating state-of-the-art single RGB image depth estimation techniques, we generate pseudo-depth representations of the frames and use distance knowledge to segment irrelevant parts of the scene. The resulting depth maps are then used as segmentation masks for the RGB frames. Experimental results on *H2O Dataset* confirm the high accuracy of the estimated pose with our method in an action recognition task. The 3D hand pose, together with information from object detection, is processed by a transformer-based action recognition network, resulting in an accuracy of 91.73%, outperforming all state-of-the-art methods. Estimations of 3D hand pose result in competitive performance with existing methods with a mean pose error of 28.66 mm. This method opens up new possibilities for employing distance information in egocentric 3D hand pose estimation without relying on depth sensors. The code is available under <https://github.com/wiktormucha/SHARP>.

**Keywords:** Egocentric · 3D hand pose · Action recognition

## 1 Introduction

In recent years, one of the growing research areas in computer vision has been egocentric vision, as evidenced by the increasing number and size of published datasets *EPIC-KITCHENS* [6], *Ego4D* [14], *H2O* [16] and release of devices like Ray-Ban Stories, Apple Vision Pro or HoloLens. One of the challenges in egocentric vision is understanding human-object interaction based on hand pose estimation and action recognition [11, 16]. The hand pose estimation task is described as the challenge of estimating the position of key points representing the joints



**Fig. 1.** Overview of our method. In the sequence of input frames  $f_1, f_2, f_3 \dots f_n$  representing the action, *SHARP* improves the estimation of the 3D hand pose  $Ph_{L,R,n}^{3D}$ . The bounding box of the manipulated objects  $Po_n^{2D}$  with their labels  $Po_l$  are retrieved using *YOLOv7* [27]. Pose information is embedded in a vector describing each frame. The sequence of vectors is processed by the transformer-based network to predict action.

of a human hand in two or three-dimensional space. Estimated positions are a valuable source of information for recognising the actions performed by a camera wearer, linking these two tasks. Egocentric action recognition research is of great importance in various domains, including augmented and virtual reality, nutritional behaviour analysis, and Active Assisted Living (AAL) technologies for lifestyle analysis [21] or assistance [17]. As AAL technologies mainly target Activities of Daily Living (ADLs) such as drinking, eating and food preparation, which are inherently manual and involve object manipulation, there’s a growing interest in research focused on hand-based action recognition.

Current work on egocentric hand-based action recognition focuses on 3D hand pose [7, 16, 26] using a single RGB camera. As a result, these studies regress  $z$  coordinate from RGB frames, which introduces complexity and results in pose prediction errors of around 40 mm [15, 16, 26] (equivalent to a 20.5% error considering an average human hand size of 18 cm), which is far from the desired performance, especially considering that publicly available datasets for egocentric hand pose are captured in a laboratory environment. Accurate pose prediction is essential for hand-based action recognition [18]. The improvement in 3D prediction could be further enhanced by the use of a depth sensor, but there’s currently no portable depth sensor on the market. Despite market availability, an additional sensor would add undesired costs due to power and processing requirements. Data growth for training and research is another constraint, as labelling key points in 3D space is difficult and requires, for example, a laboratory multi-view camera setup [16, 22]. All these circumstances create a need and motivate our research to explore new techniques and solutions to improve egocentric 3D pose estimation based on RGB images only.



Our study proposes the use of pseudo-depth images, depth images generated from a single RGB image using state-of-the-art depth estimation methods. The resulting distance representation of the scene does not contain real depth values, but it allows for the removal of non-relevant information in the scene depending on the distance. In an egocentric perspective, human arms have a constant maximum distance from the camera because the camera is mounted in a fixed position on the human body. This characteristic allows for the removal of the values representing the parts of the scene beyond this distance, leaving the input image of a hand pose estimation network with only hands and manipulated objects visible. We call this process Segmentation of Hands and Arms by Range using Pseudo-depth (SHARP). This solution requires no additional sensors; it can be applied to any RGB input data; no additional training of the depth estimation model is required; and compared to background subtraction based on image sequences, only a single RGB image is required. These advantages are confirmed by a performance improvement of 7 mm, reducing the mean pose error from 35.48 mm to 28.66 mm from the baseline. The overview of the method is presented in Fig. 1. Our contribution can be listed as follows:

- Inspired by superior egocentric hand pose estimation in 2D over other methods, we extend the state-of-the-art *EffHandEgoNet* [18] to 3D pose estimation, resulting in a new architecture called *EffHandEgoNet3D*.
- On the top of *EffHandEgoNet3D* we propose *SHARP module*, a novel idea for egocentric scene segmentation to improve hand-object interaction understanding. A state-of-the-art depth estimation model is used to generate a pseudo-depth scene representation. Furthermore, the generated distance knowledge is used to remove irrelevant information in the scene with a fixed distance over the range of the human arms, resulting in the preservation of the human arms and the interacting object. *SHARP* requires no additional training and can be applied to any egocentric RGB data. The proposed architecture outperforms several state-of-the-art studies, achieving a mean error of 28.66 mm on the *H2O Dataset*.
- We implement an action recognition network based on a transformer architecture. It uses previously estimated 3D hand pose and 2D object detection information as input. The network outperforms the state-of-the-art on the *H2O Dataset*, including methods that use more information e.g. 6D object pose, reaching 91.73% action recognition accuracy.
- We present extensive experiments and ablations performed on *H2O Dataset*, showing the influence of the proposed scene segmentation method on the performance of 3D hand pose estimation in the egocentric perspective.

The structure of the paper is as follows: In Sect. 2, we review related research on egocentric 3D hand keypoint estimation, hand-based action recognition, and depth estimation using a single RGB image, and identify opportunities for improvement. Section 3 details our approach and its implementation. Our evaluation and experimental results are presented in Sect. 4. Finally, Sect. 5 concludes the study, summarising its main findings and limitations.

## 2 Related Work

***Egocentric Hand Pose Estimation.*** Hand pose estimation in egocentric vision faces challenges such as self-occlusion, limited field of view, and diverse perspectives, which hinder effective generalisation. Some approaches overcome these obstacles by using RGB-D sensors [11, 19, 31]. However, the adoption of depth sensors is hampered by limited market availability, directing towards self-made solutions and increasing computing and power costs. Due to device limitations, several studies estimate 3D keypoints from RGB images only by using neural networks that estimate the z coordinate representing depth along x and y, followed by a conversion from 2D to 3D space using intrinsic camera parameters [16, 26]. For example, Tekin et al. [26] compute the 3D pose of a hand directly from a single RGB image using a convolutional neural network (CNN) that outputs a 3D grid with the probability of target pose values in each cell. Similarly, Kwon et al. [16] extend this approach to estimate poses for both hands. However, these methods report a mean end-point error (EPE) of 37 mm for hand pose estimation in the *H2O dataset*, suggesting room for improvement given the average human hand size of 18 cm. Cho et al. [5] use CNNs with transformer-based networks for 3D pose reconstruction on a frame-by-frame basis, while Wen et al. [30] propose a sequence-based approach for depth reconstruction that addresses occlusion challenges.

***Egocentric Action Recognition.*** A common strategy for action recognition involves the joint processing of hand and object information. Cartas et al. [3] proposes CNN-based object detectors to estimate the positions of primary regions (hands) and secondary regions (objects). Temporal information from these regions is then processed by a Long Short-Term Memory (LSTM) network. Nguyen et al. [20] Transition from bounding box information to 2D skeletons of a single hand estimated by CNN from RGB and depth images. The joints of these skeletons are aggregated using spatial and temporal Gaussian aggregation, and action recognition is performed using a learnable Symmetric Positive Definite (SPD) matrix. With the rise of 3D-based hand pose estimation algorithms, the scientific community has increasingly focused on egocentric action understanding using 3D information [7, 16, 26]. Tekin et al. [26] estimate 3D hand and object poses from a single RGB frame using a CNN, embedding temporal information to predict action classes using an LSTM. Other techniques use graph networks, such as Das et al. [7], who present a spatio-temporal graph CNN architecture that describes finger motion using separate subgraphs. Kwon et al. [16] construct sub-graphs for each hand and object, which are merged into a multigraph model, allowing learning of interactions between these components. Wen et al. [30] use a transformer-based model with estimated 3D hand pose and object label input. Cho et al. [5] enrich the transformer inputs with object pose and hand-object contact information. However, these studies do not make use of depth data. Instead, they estimate points in 3D space using neural networks and intrinsic camera parameters [5, 16, 26, 30].

**Depth Estimation from Single RGB Image.** Recent advances in depth estimation have relied on CNNs for direct regression of scene depth from input images [9]. These methods often struggle to generalise to unconstrained scenes due to the limited diversity and size of the training data. Garg et al. [12] proposed the use of calibrated stereo cameras for self-supervision, which simplifies data acquisition but maintains constraints on specific data regimes. Despite subsequent self-supervised approaches [13], challenges remain, particularly for dynamic scenes. Efforts to overcome these limitations include crowd-sourced annotation of ordinal relationships [4], but existing datasets are often biased or lack dynamic objects, making it difficult to generalise to less constrained environments. In response, Ranftl et al. [24] propose tools for mixing multiple datasets, even with incompatible annotations. Their approach incorporates a robust training objective, principled multi-objective learning, and emphasises pre-training of encoders on ancillary tasks. By training on five different sources, including a rich dataset of 3D movies, they outperform state-of-the-art depth estimation models in zero-shot cross-dataset performance. As an extension of this work, Ranftl et al. [23] present *DPT-Hybrid* and *DPT-Large* architectures enhanced with dense prediction transformers, which use vision transformers instead of CNNs, further improving the performance of depth estimation.

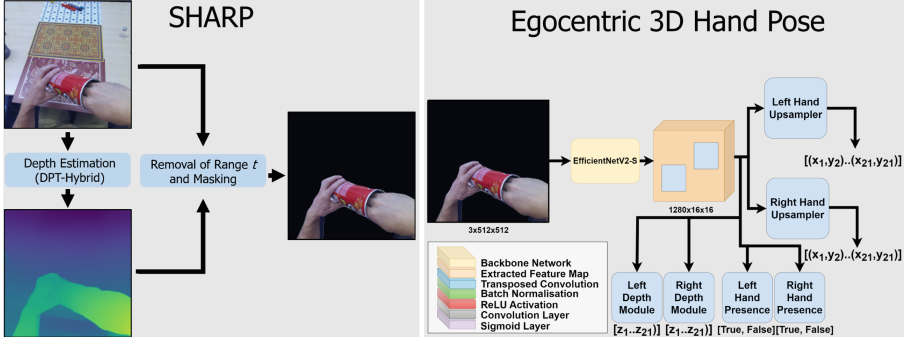
**What distinguishes our work** from other studies of egocentric 3D hand pose is the use of a depth estimation that we incorporate into *SHARP* module. Using state-of-the-art single RGB image depth estimation techniques, we generate a pseudo-depth representation of the image without any additional equipment. Knowing that the distance of the human arms from the camera in an egocentric view is constant, we then use this generated depth image to segment irrelevant information from the scene using a fixed distance threshold, thereby unifying the dataset for hand pose estimation. This methodology ensures that the hand pose estimation model only considers hands and manipulated objects, thereby increasing accuracy and efficiency, and can be applied to any RGB dataset.

### 3 Egocentric 3D Hand Pose Estimation and Action Recognition Enforced With Pseudo Depth

The study considers the tasks of egocentric 3D hand pose estimation and action recognition. These two tasks are correlated but significantly different, so the methodology is described separately for each.

#### 3.1 Egocentric 3D Hand Pose with Pseudo-depth Segmentation

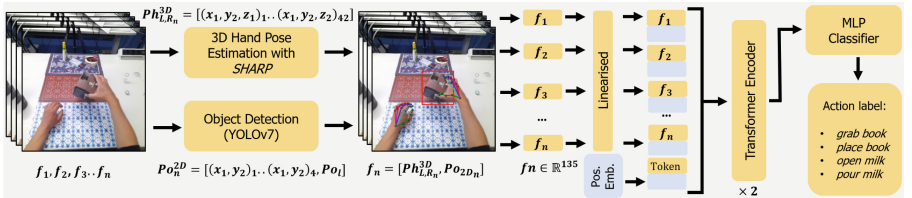
In the first stage, each RGB frame  $f_n$  undergoes processing with *SHARP* module which consists of a depth estimation model *DPT-Hybrid* [23], yielding a pseudo-depth representation  $I_n^D$  of the frame  $f_n$ . This pseudo-depth map is then normalised with its maximum value  $\max(I_n^D)$ . As human arms have a constant maximum range we utilise this characteristic. Subsequently, a fixed threshold  $t$



**Fig. 2.** Overview of the proposed egocentric 3D hand pose estimation method. First, the RGB image is processed with the *SHARP* module. Within *SHARP*, the pseudo-depth image is generated using the *DPT-Hybrid*. This distance representation is used to remove irrelevant scene information using a fixed threshold of the human arm range  $t$ . Secondly, the *SHARP* output is passed through a 3D hand pose estimation network.

is applied to the pseudo depth map  $I_n^D$  to remove the non-relevant scene part. The resultant depth map, devoid of background interference, serves as a segmentation mask for the  $f_n$ . Segmentation of  $f_n$  with  $I_n^D$  results in  $I_n^{SEG}$  where the RGB image contains only human arms and a manipulated object.

The processed  $I_n^{SEG} \in \mathbb{R}^{3 \times w \times h}$ ,  $w, h = 512$  is then inputted into a 3D hand pose estimation network, named *EffHandEgoNet3D*, which is an extension of the state-of-the-art 2D egocentric hand pose network, *EffHandEgoNet* [18], tailored for 3D estimation. *EffHandEgoNet3D* comprises an *EfficientNetV2-S* [25] backbone which extract feature map representation of  $I_n^{SEG}$   $F_M \in \mathbb{R}^{1280 \times 16 \times 16}$ . Extracted feature map  $F_M$  is handed to two independent upsamplers for each of the hands and  $MLP_{L,R}^Z$  estimating keypoints' depth. Despite pose estimation, the handness modules responsible for predicting each hand's presence



**Fig. 3.** Our action recognition procedure. From the sequence of frames  $f_1, f_2, f_3 \dots f_n$  the hand pose  $Ph_{L,R}^{3D}$  is estimated with *SHARP* and *EffHandEgoNet3D* model and the object pose  $Po_n^{2D}$ ,  $Po_1$  is extracted with *YOLOv7* [27]. Each sequence frame  $f_n$  is linearised and positional embedding and classification tokens are added. Next, this sequence is passed to a transformer encoder [8] repeated  $\times 2$  times, which embeds the temporal information. Finally, the MLP predicts one of the 36 action labels.

$h_L, h_R \in \mathbb{R}^2$  are built from another  $MLP_{L,R}^H$ . The upsamplers consist of three transposed convolutions with batch normalisation and ReLU activation except the last layer followed by a pointwise convolution. Output results are heatmaps  $H_{L,R} \in \mathbb{R}^{J \times w \times h}$  where each cell represents the probability of joint  $J$  occurrence for each hand. In the next step they are transformed into  $P_{L,R}^{2D}$  and concatenated with estimated corresponding  $z$  values resulting in  $P_{L,R}^{2.5D}$ . The final step utilises camera intrinsic parameters to transform  $P_{L,R}^{2.5D}$  using the pinhole camera model to camera space resulting in  $P_{L,R}^{3D}$ . The overview of the complete method is visible in Fig. 2.

### 3.2 Egocentric Action Recognition Based on 3D Hand Pose

We perform egocentric action recognition from image sequences using estimated 3D hand pose and 2D information about interacting object. The actions considered in this study are those in which humans manipulate objects with one or both hands, such as *pouring milk* or *opening a bottle*. An overview of the pipeline is shown in Fig. 3. It consists of three distinct components: object detection, 3D hand pose estimation, and finally action recognition using a transformer encoder and a classification MLP. The architecture improves egocentric action recognition based on the 2D hand pose introduced in *EffHandEgoNet* study [18]. The first step in the pipeline is object detection, which is carried out employing the pre-trained *YOLOv7* network [27]. In each frame, denoted as  $f_n$ , the interacting object is represented by  $P_{O2D}(x, y) \in \mathbb{R}^{4 \times 2}$ , where each point corresponds to the corners of its bounding box. Additionally,  $P_{O_l} \in \mathbb{R}^1$  represents object’s label.

The representation of each action sequence consists of frames  $[f_1, f_2, f_3, \dots, f_n]$ , where  $n \in [1..N]$  and  $N = 20$  following [18]. These frames embed flattened poses of hands  $Ph_{L,R}^{3D}$  and object  $P_{O2D}, P_{O_l}$ . If fewer than  $N$  frames represent an action, zero padding is applied, while actions longer than  $N$  frames are sub-sampled. The input vector  $V_{seq}$  is a concatenation of frames  $f_n \in \mathbb{R}^{135}$ .

$$f_n = [Ph_L^{3D}, Ph_R^{3D}, P_{O2D}, P_{O_l}] \quad (1)$$

$$V_{seq} = [f_1, f_2 \dots f_n], n \in [1..N] \quad (2)$$

The sequence vector representing an action  $V_{seq}$  is processed to embed temporal information with a transformer encoder block following [18]. First,  $V_{seq}$  is linearised using a fully connected layer to  $x_{lin}$ . The resulting  $x_{lin}$  is combined with a classification token and a positional embedding. The embedded sequence is passed to MLP for classifying the action.

## 4 Experiments

### 4.1 Datasets

In this evaluation, we focus exclusively on the *H2O Dataset* [16] due to its suitability for our research objectives. This dataset captures human actions from

an egocentric perspective, providing labels for action recognition and 3D hand pose of both hands. At the time of this study, there are only two other publicly available datasets with similar characteristics required for our study, such as *AssemblyHands* [22] and *HoloAssist* [29]. While *HoloAssist* is potentially valuable, the hand pose labels have not yet been released. *AssemblyHands* is excluded due to images captured by infrared cameras, which are incompatible with the *DPT-Hybrid* depth estimation model designed for RGB input.

***H2O Dataset*** is a comprehensive resource for analysing hand-based actions and object interactions involving two hands. It includes multi-view RGB-D images annotated with action labels covering 36 different classes derived from verb and object labels. It also includes 3D poses for both hands, resulting in  $j = 2 \times 21$  points, and 6D poses and meshes for the manipulated objects. Ground truth camera poses and scene point clouds further enrich the dataset. The actions captured in the dataset were performed by four people. For both the action recognition and hand pose estimation tasks, the dataset provides training, validation and test subsets. The action recognition subset contains 569 clips for training, 122 for validation and 242 for testing.

## 4.2 Metrics

To evaluate the hand pose estimation and compare our work with the state of the art, we calculate the Mean Per Joint Position Error (MPJPE) in millimetres over 21 keypoints  $J$  representing the human hand. This error metric quantifies the Euclidean distance between the predicted and ground truth values. For action recognition, we use the top-1 accuracy measure, where the model’s prediction must exactly match the expected ground truth to be considered accurate.

## 4.3 Experiment Setup

For both learning processes, each run is repeated three times to reduce the effect of random initialisation of the network, and mean results with standard deviations are reported.

***3D Hand Pose Estimation*** is trained and evaluated on *H2O Dataset*. The optimisation is done using Stochastic Gradient Descent (SGD) over the summarised loss function including Intersection over Union (IoU) for each upsampler and  $L1$  loss for predicted corresponding depth values. The process starts with a learning rate  $l_r = 0.1$  and momentum equal to  $m = 0.9$ . Over time  $l_r$  is reduced by  $\alpha = 0.5$  every  $10^{th}$  epoch starting from the  $50^{th}$  epoch. The data is augmented with random cropping, horizontal flipping, vertical flipping, resizing, rotating and blurring. The batch size is equal to  $b_s = 32$ . Model weights are saved for the smallest MPJPE in the validation subset.

***Action Recognition*** module requires object detection. For this, we fine-tune YOLOv7 on the *H2O Dataset* using the open-source strategy reported by the authors. The training of the action recognition includes the augmentation of

the sequence vectors with keypoints using random rotation and an additional strategy with random masking of either the hand, the object positions or the label. This is done by setting the corresponding values of the hand or object in the frame  $f_n$  to zero. We follow [16, 26, 32] and use given poses in training. Input sequence frames are randomly sub-sampled during training and uniformly sub-sampled for validation and testing. Models are trained with a batch size  $b_s = 64$ , AdamW optimiser, cross-entropy loss function, and a learning rate  $l_r = 0.001$  reduced by a factor of 0.5 every 200 epochs after 500 epochs. Hyperparameters and augmentations are selected based on the best-performing set in the validation subset. Weights are stored for best validation accuracy.

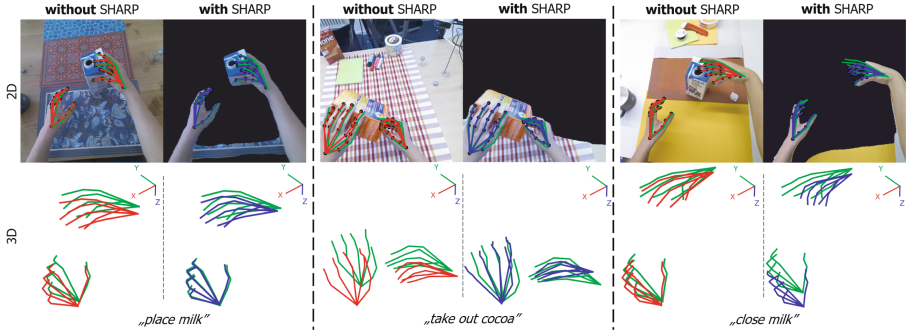
#### 4.4 Comparison with State of the Art

Our architecture with *SHARP* gives an average MPJPE in hand pose of  $29.61 \pm 0.71$  mm in three consecutive runs with the best run MPJPE equal to 28.66 mm. The qualitative results shown in Fig. 4 confirm the improvement in 3D hand pose estimation when using *SHARP*, but also show that *SHARP* can lead to a degradation in performance if too much information is reduced from the scene. Further, we employ the estimated 3D hand pose using *SHARP* in the proposed action recognition architecture. It yields an average of  $90.90\% \pm 0.67$  over three runs, with the best model yielding an accuracy of 91.73%. Comparison with state of the art for egocentric 3D hand pose estimation is presented in Table 1. Table 2 presents a comparison of state-of-the-art action recognition methods and their results on the *H2O Dataset* reported by the authors. To ensure a fair comparison, the table provides details regarding the inputs of the action recognition modules. For both tasks, we follow other studies [1, 5, 18, 26, 30] and report our best results.

We measure the inference times of our methods for the hand pose estimation task for a single frame and for a complete action recognition pipeline for a single action. The evaluation is performed by averaging the inference times over 1000 trials on the NVIDIA GeForce RTX3090 GPU for reliability. The results are shown in Fig. 5, where the upper part shows the hand pose performance and the lower part shows the action recognition. Our methods are compared with HTT

**Table 1.** Results of 3D hand pose estimation provided in *mm* in camera space.

Method	Year	MPJPE Left ↓	MPJPE Right ↓	MPJPE Both ↓
LPC [15]	2020	39.56	41.87	40.72
H+O [26]	2019	41.42	38.86	40.14
H2O [16]	2021	41.45	37.21	39.33
HTT [30]	2023	35.02	35.63	35.33
H2OTR [5]	2023	24.40	25.80	<b>25.10</b>
THOR-Net [1]	2023	36.80	36.50	36.65
<b>Ours</b>	Now	30.31	27.02	28.66



**Fig. 4.** Qualitative results of our method in 2D and 3D space. Green skeletons represent the **ground truth hand pose**, red estimations **without SHARP** and blue estimations **with SHARP**. Images are annotated with a predicted action label for the represented sequences. Two examples from the left show that *SHARP* improves 3D pose estimation. On the right, the 3D error increases as *SHARP* partially loses the right hand. (Color figure online)

[30] and H2OTR [5] as they are the only open-source implementations that allow such a comparison on the *H2O Dataset* at the time of this study.

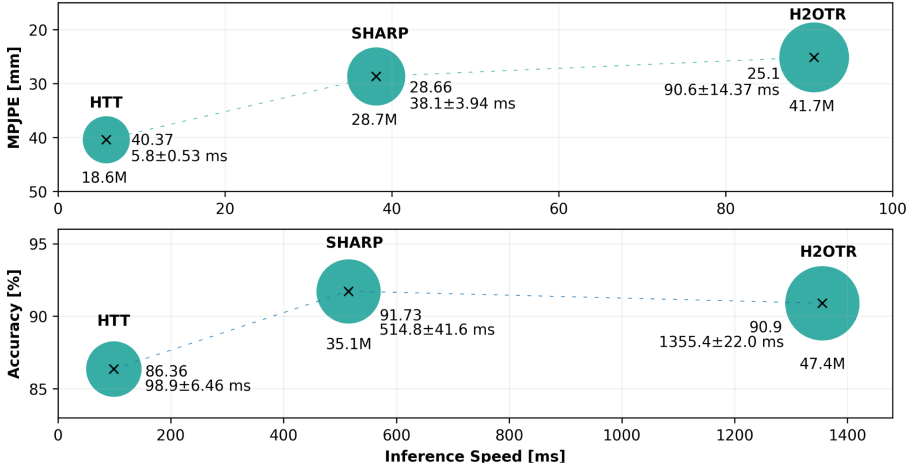
*SHARP* estimates the egocentric 3D hand pose with the second best result, being faster  $\approx \times 2.4$  than the best H2OTR [5] with 13M fewer parameters and only a 3mm performance penalty. Our action recognition outperforms all state-of-the-art methods and infers  $\approx \times 2.6$  faster with 12M fewer parameters than the second best H2OTR [5].

#### 4.5 Ablation Studies

To further evaluate our approach, we conduct extensive ablation studies. All experiments are performed with a fixed number of seeds to ensure reproducibility by eliminating the effect of random initialisation.

***The Range of Human Arms in Training.*** The most important part of our architecture is the pseudo-depth-based distance segmentation, which aims to remove irrelevant information from the processed scene, except for the human hands and the manipulated object. It raises the key question of what value of distance should be used as the threshold  $t$ . In the case of pseudo depth obtained with *DPT-Hybrid*, the depth values are normalised, where  $t \in \langle 0, 1 \rangle$ . To select  $t$ , we first observe the dataset samples and choose values that lead to the preservation of hands and objects only. However, as it is based on estimation, the behaviour is not the same for all samples for the same  $t$  and none of these values can be considered good without being proven with performance. In the second step, we search for the best performance by retraining the architecture for each of these  $t \in \{0.35, 0.39, 0.43, 0.47, 0.51\}$ . The results highlight  $t = 0.47$  as the highest performance value and we observe the performance decrease above and





**Fig. 5.** Inference time for 3D hand pose estimation per single frame and action recognition accuracy per single action of state-of-the-art methods on *H2O Dataset*. Each method is visualised as a circle whose size represents the number of trainable parameters. *SHARP* inference is  $\approx \times 2.5$  faster than *H2OTR* [5] with better action recognition.

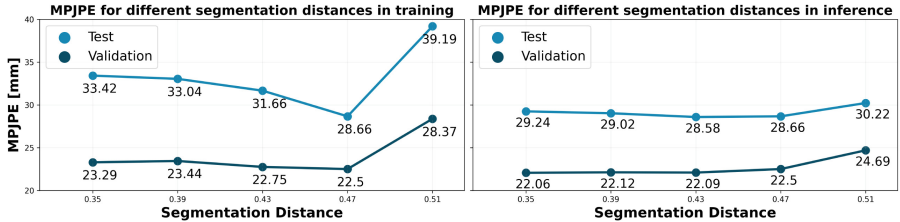
below this value, proving the usability of the proposed method. All results are presented in the left sub-figure of Fig. 6.

**The Range of Human Arms in Inference.** Following the choice of  $t$  in training, we examine the choice of  $t$  in testing for the best-performing model with  $t = 0.47$  in training. We run tests for  $t \in \{0.35, 0.39, 0.43, 0.47, 0.51\}$ . All results are shown in the right subplot of Fig. 6. The effect of  $t$  is significantly lower than in training and does not affect performance much.

**Pseudo-depth-Based SHARP Module.** We evaluate *SHARP*'s impact on the egocentric 3D hand pose estimation performance. The proposed architecture is retrained according to the previously described process without the *SHARP* module, using only unsegmented RGB images representing the full scene. The network reduced by the *SHARP* module in a fixed seed run achieves an MPJPE of 35.48 mm compared to 28.66 mm obtained with *SHARP*. The result is referenced in Table 3 as *Ablation I*. The process is repeated three times to reduce the random effect of network initialisation and to strengthen the justification of the idea. The average of the three runs without the *SHARP* module is  $35.34 \pm 0.17$ , while with *SHARP*, the performance improves to  $29.61 \pm 0.71$  mm, demonstrating the high importance of the proposed architecture.

**Table 2.** Results in accuracy of action recognition methods on *H2O Dataset*. Inputs of methods are: *Img* stands for semantic features extracted from an image using CNN network, *Hand Pose* and *Obj Pose* stand for pose information type for hands and objects, and *Obj Label* stands for object label. Results origin from referenced studies.

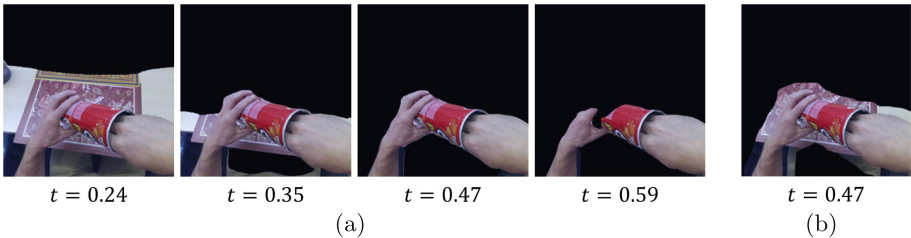
Method	Year	Img	Hand Pose	Obj Pose	Obj Label	Acc. $\uparrow$
C2D [28]	2018	✓	✗	✗	✗	70.66
I3D [2]	2017	✓	✗	✗	✗	75.21
SlowFast [10]	2019	✓	✗	✗	✗	77.69
H+O [26]	2019	✗	3D	6D	✓	68.88
ST-GCN [32]	2018	✗	3D	6D	✓	73.86
TA-GCN [16]	2021	✗	3D	6D	✓	79.25
HTT [30]	2023	✓	3D	✗	✓	86.36
H2OTR [5]	2023	✗	3D	6D	✓	90.90
EffHandEgoNet [18]	2024	✗	2D	2D	✓	91.32
<b>Ours</b>	Now	✗	3D	2D	✓	<b>91.73</b>



**Fig. 6.** Figures showing the results of the 3D hand pose estimation error in MPJPE as a function of the segmentation threshold  $t$ . The left figure shows the performance with different thresholds used for training and the right figure shows the performance for the best trained model with  $t = 0.47$  and different  $t$  during inference.

**Oracle Depth-Based SHARP Module.** *SHARP* uses the state-of-the-art depth estimation network *DPT-Hybrid*. Like any deep learning architecture, this model is prone to errors. On the other hand, with progress in architecture development, depth estimation networks will improve in the future, leading to an improvement in the performance of our method. To highlight this potential, we retrain the network with an oracle ground truth depth image provided in the *H2O Dataset*. The depth image represents the distance in mm from a camera. For this reason, we choose  $t = 700$  mm. The results are superior, achieving an MPJPE of 25.09 mm, better than any state-of-the-art method at the time of this study. The experiment is referred as *Ablation II* in Table 3. This performance demonstrates the potential of our approach when fed with less noisy pseudo-depth data.

**De-sharpening of Segmentation Mask.** The segmentation mask, derived from a pseudo-depth scene representation, consists of sharp edges surrounding the human arms and the manipulated object, based on a distance. Depth estimation is prone to error, and in some scenes, this sharp-edge segmentation leads to the loss of parts of the image that represent relevant information, e.g. human hand. This negative effect can be reduced in two ways, by changing the segmentation threshold as shown in Fig. 7(a) or by de-sharpening the edges. The effect of the de-sharpening process is presented in Fig. 7(b). In this ablation, we observe the effect of edge de-sharpening by blurring the mask derived from the pseudo-depth scene representation. Performance drops to 37.25 mm, highlighting the usefulness of the *SHARP* module only with accurate masking.



**Fig. 7.** On the left, frame processed with *SHARP* and different values of  $t$ . On the right, the same frame processed with *SHARP*,  $t = 0.47$  and with de-sharpening applied.

**Table 3.** Results of ablations studies with different depth image types used in *SHARP*. All results provided in *mm* in camera space for left, right and both hands.

	Depth	MPJPE Left ↓	MPJPE Right ↓	MPJPE Both ↓
Ours	Estimated	30.31	27.02	28.66
Ablation I	✗	32.95	38.01	35.48
Ablation II	Ground Truth	21.31	28.86	25.09
Ablation III	Est.+De-sharpen	39.49	35.01	37.25

## 5 Conclusion

In this study, a 3D hand pose estimation model has been developed for the egocentric perspective. The novelty of the proposed architecture lies in the *SHARP* module, which uses pseudo-depth scene representation obtained through a monocular depth estimation model. Thanks to the characteristic of a fixed camera to a user in the egocentric perspective and a constant range of human arms, the distance information is used to remove irrelevant information from the scene. Experiments with our network showed an improvement in performance of 7 mm

in the MPJPE metric when using *SHARP*, with the best result of MPJPE equal to 28.66 mm placing as the second best result on the *H2O Dataset*. The further potential of the *SHARP* module was confirmed with the use of the ground truth depth image, resulting in the best result of all state-of-the-art methods equal to 25.09 mm. Furthermore, estimated 3D hand poses were used alongside object detection as input for the action recognition model, where each frame is described by a vector containing the 3D hand pose and the object bounding box, and their sequence is embedded using a transformer-based network. The results obtained on *H2O Dataset*, which includes actions where one hand or two hands interact with objects, resulted in 91.73% accuracy, outperforming the state-of-the-art.

Our study shows that using pseudo depth to remove irrelevant information in the egocentric scene with current state-of-the-art monocular depth estimation methods improves 3D hand pose performance. The quality of pseudo depth correlates with pose estimation error and requires a sharp and accurate representation of human hands in the scene. In the future, with the advancement of depth estimation networks, this approach has a chance to improve hand pose estimation tasks further, leading to more accurate action recognition.

**Acknowledgements.** Part of this work was conducted during Wiktor’s research secondment at the University of Bristol within the Machine Learning and Computer Vision Research Group (MaVi). We thank the group for their support and resources. This research was supported by VisuAAL ITN H2020 (grant agreement no. 861091) and the Austrian Research Promotion Agency (grant agreement no. 49450173).

## References

1. Aboukhadra, A., Malik, J., Elhayek, A., Robertini, N., Stricker, D.: THOR-Net: end-to-end graformer-based realistic two hands and object reconstruction with self-supervision. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1001–1010 (2023). <https://doi.org/10.1109/WACV56688.2023.00106>
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017). <https://doi.org/10.1109/CVPR.2017.502>
3. Cartas, A., Radeva, P., Dimiccoli, M.: Contextually driven first-person action recognition from videos. In: Presentation at EPIC@ ICCV2017 Workshop, p. 8 (2017)
4. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in Neural Information Processing Systems 29 (2016)
5. Cho, H., Kim, C., Kim, J., Lee, S., Ismayilzada, E., Baek, S.: Transformer-based unified recognition of two hands manipulating objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4769–4778 (2023). <https://doi.org/10.1109/CVPR52729.2023.00462>
6. Damen, D., et al.: Scaling egocentric vision: the EPIC-KITCHENS dataset. In: European Conference on Computer Vision (ECCV) (2018). [https://doi.org/10.1007/978-3-030-01225-0\\_44](https://doi.org/10.1007/978-3-030-01225-0_44)

7. Das, P., Ortega, A.: Symmetric sub-graph spatio-temporal graph convolution and its application in complex activity recognition. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3215–3219. IEEE (2021). <https://doi.org/10.1109/ICASSP39728.2021.9413833>
8. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems 27 (2014)
10. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6202–6211 (2019). <https://doi.org/10.1109/ICCV.2019.00630>
11. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 409–419 (2018). <https://doi.org/10.1109/CVPR.2018.00050>
12. Garg, R., B.G., V.K., Carneiro, G., Reid, I.: Unsupervised CNN for single view depth estimation: geometry to the rescue. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 740–756. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46484-8\\_45](https://doi.org/10.1007/978-3-319-46484-8_45)
13. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3828–3838 (2019). <https://doi.org/10.1109/ICCV.2019.00393>
14. Grauman, K., et al.: Ego4D: around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18995–19012 (2022). <https://doi.org/10.1109/CVPR52688.2022.01842>
15. Hasson, Y., Tekin, B., Bogo, F., Laptev, I., Pollefeys, M., Schmid, C.: Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 571–580 (2020). <https://doi.org/10.1109/CVPR42600.2020.00065>
16. Kwon, T., Tekin, B., Stühmer, J., Bogo, F., Pollefeys, M.: H2O: two hands manipulating objects for first person interaction recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10138–10148 (2021). <https://doi.org/10.1109/iccv48922.2021.00998>
17. Mucha, W., Cuconasu, F., Etori, N.A., Kalokyri, V., Trappolini, G.: TEXT2TASTE: a versatile egocentric vision system for intelligent reading assistance using large language model. In: Computers Helping People with Special Needs, pp. 285–291. Springer, Cham (2024). [https://doi.org/10.1007/978-3-031-62849-8\\_35](https://doi.org/10.1007/978-3-031-62849-8_35)
18. Mucha, W., Kampel, M.: In my perspective, in my hands: accurate egocentric 2D hand pose and action recognition. In: 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1–9 (2024). <https://doi.org/10.1109/FG59268.2024.10582035>
19. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1154–1163 (2017). <https://doi.org/10.1109/CVPR.2019.01231>

20. Nguyen, X.S., Brun, L., Lézoray, O., Bouglex, S.: A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12036–12045 (2019). <https://doi.org/10.1109/CVPR.2019.01231>
21. Núñez-Marcos, A., Azkune, G., Arganda-Carreras, I.: Egocentric vision-based action recognition: a survey. *Neurocomputing* **472**, 175–197 (2022). <https://doi.org/10.1016/j.neucom.2021.11.081>
22. Ohkawa, T., He, K., Sener, F., Hodan, T., Tran, L., Keskin, C.: AssemblyHands: towards egocentric activity understanding via 3D hand pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12999–13008 (2023). <https://doi.org/10.1109/CVPR52729.2023.01249>
23. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12179–12188 (2021). <https://doi.org/10.1109/ICCV48922.2021.01196>
24. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(3), 1623–1637 (2020). <https://doi.org/10.1109/TPAMI.2020.3019967>
25. Tan, M., Le, Q.: EfficientNetV2: smaller models and faster training. In: International Conference on Machine Learning, pp. 10096–10106. PMLR (2021)
26. Tekin, B., Bogo, F., Pollefeys, M.: H+O: unified egocentric recognition of 3D hand-object poses and interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4511–4520 (2019). <https://doi.org/10.1109/CVPR.2019.00464>
27. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464–7475 (2023). <https://doi.org/10.48550/arXiv.2207.02696>
28. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7794–7803 (2018). <https://doi.org/10.1109/CVPR.2018.00813>
29. Wang, X., et al.: HoloAssist: an egocentric human interaction dataset for interactive AI assistants in the real world. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 20270–20281 (2023). <https://doi.org/10.1109/ICCV51070.2023.01854>
30. Wen, Y., Pan, H., Yang, L., Pan, J., Komura, T., Wang, W.: Hierarchical temporal transformer for 3D hand pose estimation and action recognition from egocentric RGB videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21243–21253 (2023). <https://doi.org/10.1109/CVPR52729.2023.02035>
31. Yamazaki, W., Ding, M., Takamatsu, J., Ogasawara, T.: Hand pose estimation and motion recognition using egocentric RGB-D video. In: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 147–152. IEEE (2017). <https://doi.org/10.1109/ROBIO.2017.8324409>
32. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018). <https://doi.org/10.1609/aaai.v32i1.12328>



# On the Generalization of WiFi-Based Person-Centric Sensing in Through-Wall Scenarios

Julian Strohmayer<sup>(✉)</sup>  and Martin Kampel 

Computer Vision Lab, TU Wien, Favoritenstr. 9/193-1, 1040 Vienna, Austria  
{julian.strohmayer,martin.kampel}@tuwien.ac.at

**Abstract.** In this work, the problem of cross-environment generalization in WiFi Channel State Information (CSI)-based localization and Human Activity Recognition (HAR) models within through-wall scenarios is addressed, highlighting an area that remains underexplored. A comprehensive evaluation is conducted to investigate the effectiveness of various methodologies, including CSI feature selection, feature scaling, dimensionality reduction, and data augmentation techniques, in improving model robustness to environmental variations. The evaluation is based on a dataset collected over three days in environments exhibiting both static and dynamic variations, featuring synchronized CSI and 3D trajectory data of human activities, which is made publicly available at <https://zenodo.org/records/10925351>. The findings reveal that plain CSI amplitude features consistently outperform other types in achieving superior generalization in through-wall scenarios. Furthermore, it is found that while dimensionality reduction techniques like PCA, ICA, and UMAP do not enhance model generalization, feature scaling and data augmentation can significantly improve both localization and HAR performance in the presence of static and dynamic environmental variations.

**Keywords:** WiFi · Through-Wall Sensing · Generalization · Localization · Activity Recognition

## 1 Introduction

In the field of person-centric sensing, WiFi has gained significant attention as a sensing modality due to its advantages over optical approaches, including cost-effectiveness, unobtrusiveness, and visual privacy protection [2, 17], as well as

This work is partly funded by the Vienna Business Agency (grant 4829418) and the Austrian security research program KIRAS of the Austrian Research Promotion Agency FFG (grant 49450173).

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_13](https://doi.org/10.1007/978-3-031-78354-8_13).

its unique ability to penetrate walls for long-range sensing in confined indoor environments [7]. This capability is particularly useful for applications such as through-wall Human Activity Recognition (HAR), which offers potential economic benefits by enabling the monitoring of human activities across vast indoor environments without the need for per-room sensor deployment [23]. Central to the advancement of WiFi-based sensing is Channel State Information (CSI), enabled by the Orthogonal Frequency-Division Multiplexing (OFDM) scheme introduced in the 802.11a standard. CSI provides a granular view of how WiFi signals, distributed across multiple subcarrier frequencies, interact with their environment, thereby capturing the dynamic variations in signal propagation caused by human activities [19]. Despite its potential in person-centric sensing applications, the sensitivity of CSI to environmental variations poses significant challenges for generalizing WiFi-based sensing systems to new environments and scenarios in practice [4, 12]. This issue is especially pronounced in through-wall scenarios, a subset of Non-Line-of-Sight (NLoS) scenarios, where the environmental impact on WiFi signals is intensified by the complex signal behavior due to reflection, diffraction, refraction phenomena, and attenuation by building materials, which are often varied and unknown [32].

**Contributions.** In response to the highlighted challenges, this work explores the impact of CSI-based feature selection, feature scaling, dimensionality reduction, and data augmentation techniques on the generalization capabilities of localization and activity recognition models within through-wall scenarios. Our comprehensive evaluation spanning three consecutive days captures both static and dynamic environmental variations, alongside long-term hardware variations, to assess the robustness of model generalization. To stimulate further research into overcoming the challenges of WiFi-based person-centric sensing across diverse environments, the dataset underlying our evaluation is made publicly available<sup>1</sup>.

## 2 Related Work

The central goal of cross-domain WiFi sensing is the generalization of models to new, unseen environments. A comprehensive survey on the state of cross-domain WiFi sensing is presented by Chen et al. [4], discussing domain-invariant feature extraction, virtual sample generation, transfer learning, few-shot learning, and big data approaches, as well as open challenges limiting practical applicability. Our work falls into the categories of domain-invariant feature extraction and virtual sample generation. One method of virtual sample generation involves spatial and temporal perturbations to CSI amplitude and phase spectrograms, as explored in [20, 24]. Furthermore, noise injection [8], or subcarrier-level dropout [25] have been proposed for enhancing model cross-domain and cross-system generalization. The exploration of derivative features such as Power Spectral Density (PSD) and first-order differences for enhancing model robustness across Line-of-Sight (LoS) and NLoS scenarios are documented in [28] and

---

<sup>1</sup> Dataset, <https://zenodo.org/records/10925351>.



[1], respectively. Furthermore, the UniFi system [14] introduces the use of person location and orientation for gesture recognition, showing progress towards environmental independence. Another system, AirFi [27], exemplifies the strategy of extracting environment-independent features to facilitate model generalization to new environments without additional data collection. In [12], a data augmentation approach based on MixUp [31] that addresses long-term variations in CSI, is proposed. Furthermore, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) have been shown to improve performance in LoS HAR scenarios [10], and Uniform Manifold Approximation and Projection (UMAP) has also been shown to enhance WiFi fingerprint-based localization performance [21, 29]. Despite these advancements, model generalization in through-wall scenarios has not been extensively studied, which leaves a significant gap in our understanding of complex signal interactions with building materials and their effect on person-centric sensing performance [32]. Our work addresses this by evaluating the effectiveness of these methods for through-wall scenarios, aiming to enhance the development of robust, generalizable WiFi-based person-centric sensing systems.

### 3 Experimental Setup

This section outlines our experimental setup, detailing the WiFi system employed, the physical layout of the evaluation environment as well as transmitter-receiver arrangement, and the procedure followed to gather the dataset underlying our evaluation.

#### 3.1 WiFi System

To collect the data required for our evaluation, we employ the WiFi system proposed in [25]. This system integrates CSI sensing and processing hardware within a compact, 3D-printed enclosure. At its core is the *ESP32-S3-DevKitC-1U*<sup>2</sup>, which features the *ESP32-S3-WROOM-1U*<sup>3</sup> microcontroller for WiFi connectivity and CSI access via ESP-IDF (See Footnote 1). In WiFi-based person-centric sensing, the built-in printed inverted F-antenna (PIFA)<sup>4</sup> of the *ESP32s* is commonly leveraged. However, its omnidirectionality and low gain of 2 dBi limit the ability to constrain the recording environment, making it susceptible to external noise [23]. To address this problem and facilitate WiFi-based person-centric sensing in long-range scenarios, the system described in [25] substitutes the PIFA with the *ALFA Network APA-M25*<sup>5</sup>, a USD 20 dual-band directional panel antenna with a 66° horizontal beam width and 8dBi gain at 2.4 GHz. This

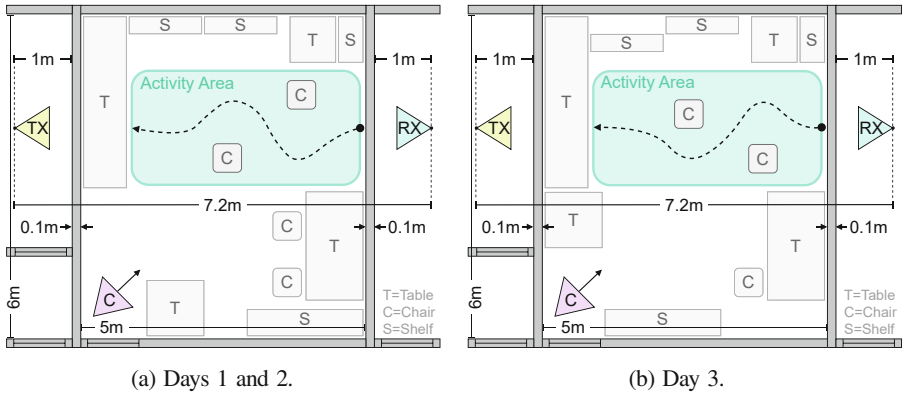
<sup>2</sup> Espressif ESP32-S3-DevKitC-1U, <https://docs.espressif.com>, accessed: 12-03-2024.

<sup>3</sup> Espressif ESP32-S3-WROOM-1U, <https://docs.espressif.com>, accessed: 12-03-2024.

<sup>4</sup> ESP32 PIFA, <https://www.ti.com>, accessed: 12-03-2024

<sup>5</sup> ALFA Network APA-M25, <https://alfa-network.eu/apa-m25>, accessed: 12-03-2024

antenna is connected to the I-PEX MHF1 connector of the *ESP32-S3-WROOM-1U* module. The resulting WiFi system, featuring high gain and directionality, is well suited for the through-wall scenarios investigated in this work. We deploy the system in a point-to-point transmitter-receiver arrangement, where one of two identical devices serves as the transmitter, sending WiFi packets at a fixed frequency of 100 Hz. The other device functions as the receiver, continually listening for WiFi packets. A WiFi connection between the transmitter and receiver is established using the wireless communication protocol, ESP-NOW<sup>6</sup>. WiFi packets are captured on the receiver unit’s integrated *Nvidia Jetson Orin Nano*<sup>7</sup> via ESP-IDF.



**Fig. 1.** Evaluation environment layout over three consecutive days: (a) layout on days 1 and 2, and (b) layout on day 3, featuring static environmental variations due to furniture rearrangement. The transmitter-receiver arrangement and the designated activity area remain fixed throughout the experiment.

### 3.2 Evaluation Environment

Figure 1a illustrates the layout of our experimental setup, which consists of a central room of interest where activities are conducted, flanked by two adjacent rooms housing the transmitter and receiver, respectively. The central room, filled with typical office furniture such as chairs, tables, and shelves, measures  $6\text{ m} \times 5\text{ m}$ . Due to physical constraints, the area designated for activities (highlighted in blue) is smaller than the room itself, covering  $4\text{ m} \times 2.5\text{ m}$ . The transmitter and receiver units are positioned in a point-to-point arrangement, spanning a distance of  $7.2\text{ m}$ , as depicted. The two walls separating the transmitter and receiver are constructed from plasterboard, each with a thickness of  $0.1\text{ m}$ . Furthermore, an additional component of our setup is a WiFi webcam within the room that

<sup>6</sup> ESP-NOW, <https://docs.espressif.com>, accessed: 12-03-2024.

<sup>7</sup> Nvidia Jetson Orin Nano, <https://developer.nvidia.com>, accessed: 12-03-2024.

streams video to the receiver which allows us to determine start and end points of activity sequences. Our experiment, aimed at capturing static, dynamic, and temporal variations in the WiFi signal and assessing their impact on model generalization, spans three consecutive days. On the first two days, the environment remains unchanged, as shown in Fig. 1a. However, on the third day, static environmental variations are introduced by rearranging large furniture pieces along with small to medium-sized objects in the room, as illustrated in Fig. 1b. Therefore, compared to day 1, days 2 and 3 exhibit both dynamic (variations in activity execution) and temporal environmental variations, with day 3 additionally featuring static environmental variations, thus posing a greater challenge from a generalization standpoint. To ensure the comparability of results across different days (especially concerning the localization problem), the positions of the transmitter and receiver remain fixed throughout the experiment.

### 3.3 Data

To assess model generalization in the presence of static, dynamic, and temporal environmental variations for person localization (3D position regression) and activity recognition, we gather a dataset of synchronized WiFi packets and trajectory data of an individual. We choose to use a single participant to eliminate variations due to physiological differences, as this well-known type of domain variation is already covered by existing datasets, and including multiple participants makes it more challenging to isolate the effects of static and dynamic environmental variations. Our focus is specifically on dynamic activity and static environmental variations in through-wall scenarios, as no existing dataset offers such a clear separation. This distinction makes our dataset a unique and valuable benchmark for developing generalization methods tailored to through-wall scenarios. The activities, categorized as *walking*, *sitting*, and *lying*, with examples provided in Fig. 2, are recorded over three consecutive days. Each day features five activity-class sequences, each lasting five minutes. During the *walking* activity, the individual moves freely within the area, avoiding chairs. For *sitting*, they alternate between two chairs, incorporating random head, arm, and leg movements to increase sample variability. Lastly, in the *lying* activity, we simulate a fall detection scenario where the person struggles on their back and slides around, as depicted in Fig. 2c. The raw data collection involves simultaneously recording WiFi packets and egocentric video using a chest-mounted camera while the individual engages in activities within the activity area, as depicted in Fig. 1a. Ground truth locations are derived from egocentric videos using ORB-SLAM3 [3]. To ensure temporal alignment between CSI and location time series, visual cues from the WiFi webcam mark the start and end of the CSI series, aiding in the removal of redundant samples. Moreover, to match the CSI’s 100 Hz sampling rate, the originally 30 Hz-sampled location time series are linearly up-sampled. The resulting dataset, the basis of our evaluation, comprises over 1.2 million samples, including WiFi CSI, 3D location, and class labels, and is made publicly available (See Footnote 1). The detailed distribution of samples across days

and activity classes is documented in Table 1. For model training, data exclusively from day 1 is used, following an 8:2 split for training and validation. Data from days 2 and 3 are reserved for testing. Notably, day 3 data contains fewer *lying* activity samples, due to the exclusion of two sequences owing to trajectory estimation errors.



**Fig. 2.** Day 1 examples of the activity classes *walking*, *sitting*, and *lying*.

**Table 1.** Distribution of data samples across days and activity classes: *walking*, *sitting*, and *lying*.

Day	<i>walking</i>	<i>sitting</i>	<i>lying</i>	all classes
1	105.288	142.242	149.803	397.333
2	110.146	181.991	181.332	473.469
3	152.504	150.338	87.935	390.777
Total	367.938	474.571	419.070	1.261.579

## 4 Methods

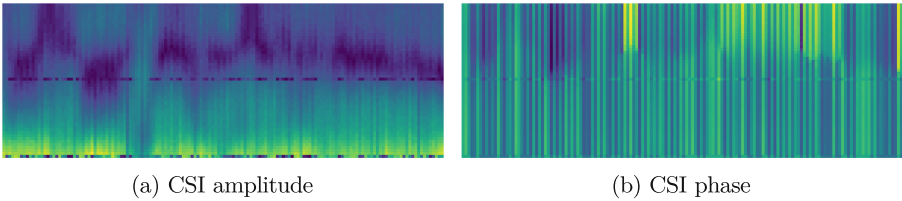
We evaluate a range of methods aimed at enhancing model generalization in through-wall scenarios. The evaluation is grounded in feature selection, including amplitude, phase, first-order differences, and PSD, to identify the most effective CSI-based features. Subsequent analyses focus on feature scaling techniques, such as max-min scaling and z-normalization, and explore dimensionality reduction methods, namely PCA, ICA, and UMAP. Additionally, we investigate perturbation-based data augmentation techniques tailored for CSI data in the image domain. The methodologies employed are detailed in the sections that follow.

#### 4.1 CSI Feature Selection

In the domain of WiFi-based person-centric sensing, the primary features extracted from CSI are the *amplitude* and *phase*. The CSI metric within OFDM systems captures the changes in amplitude and phase across subcarrier frequencies of a signal transmitted between a transmitter and a receiver. Following the notation in [10], the estimated received signal vector  $y$  is expressed as  $y = \mathbb{H}x + \eta$ , where  $\mathbb{H}$  is the CSI matrix,  $x$  is the transmitted signal vector, and  $\eta \sim \mathcal{N}(\mu, \Sigma)$  represents an additive Gaussian noise vector. The components of  $\mathbb{H}$  are complex numbers  $h_i = A_i e^{j\phi_i}$ , indicating the Channel Frequency Response (CFR) for the  $i$ th subcarrier, where amplitude  $A_i$  and phase  $\phi_i$  are computed using the real  $\mathcal{R}(h_i)$  and imaginary  $\mathcal{I}(h_i)$  parts:

$$A_i = \sqrt{(\mathcal{I}(h_i))^2 + (\mathcal{R}(h_i))^2} \quad (1)$$

$$\phi_i = \text{atan2}(\mathcal{I}(h_i), \mathcal{R}(h_i)). \quad (2)$$



**Fig. 3.** Examples of (a) CSI amplitude,  $\mathcal{A}[t]$ , and (b) phase matrices,  $\mathcal{P}[t]$ , for a person walking in a through-wall scenario are presented. These images display the amplitude and phase of 52 L-LTF subcarriers across a time span of approximately 1.5 s, equivalent to 150 WiFi packets.

Given that human activities typically extend over certain periods of time, CSI from a selection of subcarriers  $S$  across a number of WiFi packets  $w$  is used, forming a  $S \times w$  CSI matrix  $\mathcal{H}[t]$ , where  $t$  is the time or packet index, as depicted in Eq. 3. From this, using the Eqs. 1 and 2, amplitude  $\mathcal{A}[t]$  and phase matrices  $\mathcal{P}[t]$  are derived which can be fed to a deep learning algorithm to perform WiFi-based person-centric sensing tasks. Exemplary visual representations of  $\mathcal{A}[t]$  and  $\mathcal{P}[t]$  are presented in Fig. 3, illustrating the temporal amplitude and phase variations induced by a person walking.

$$\mathcal{H}[t] = \begin{bmatrix} h_1[t - \lfloor \frac{w}{2} \rfloor] & h_1[t - \lfloor \frac{w}{2} \rfloor + 1] & \cdots & h_1[t + \lfloor \frac{w}{2} \rfloor] \\ h_2[t - \lfloor \frac{w}{2} \rfloor] & h_2[t - \lfloor \frac{w}{2} \rfloor + 1] & \cdots & h_2[t + \lfloor \frac{w}{2} \rfloor] \\ \vdots & \vdots & \ddots & \vdots \\ h_S[t - \lfloor \frac{w}{2} \rfloor] & h_S[t - \lfloor \frac{w}{2} \rfloor + 1] & \cdots & h_S[t + \lfloor \frac{w}{2} \rfloor] \end{bmatrix} \quad (3)$$

As environmental variations impact these amplitude and phase features, models trained on such data may face generalization problems across different environments. To address this, *first-order difference* (temporal difference) features based on amplitude or phase are proposed [1,6], capturing the change between consecutive time steps rather than absolute values, thus potentially enhancing environmental generalization. First-order difference features are defined as follows:

$$h_{\Delta}[t] = h[t] - h[t - 1]. \quad (4)$$

Applying Eq.4 to the CSI time series  $h$  yields the first-order difference time series  $h_{\Delta}$ , from which  $\mathcal{H}_{\Delta}[t]$ , and subsequently the first-order difference amplitude  $\mathcal{A}_{\Delta}[t]$  and phase matrices  $\mathcal{P}_{\Delta}[t]$ , can be extracted. Finally, *Power Spectral Density* (PSD), which translates the CSI time-series data  $h$  into the frequency domain via the Fast Fourier Transform (FFT), offers another alternative for feature extraction. The PSD is computed for each subcarrier across a window size  $w$ , leading to the PSD matrix  $\mathcal{PSD}[t]$ :

$$h_{PSD}[t] = \frac{|\text{FFT}(h[t])|^2}{w}. \quad (5)$$

## 4.2 Feature Scaling

*Max-min scaling* (or min-max normalization) is a feature scaling technique used in machine learning to normalize the range of independent variables or features of data. It scales the features to a fixed range, typically 0 to 1, by subtracting the minimum value of the feature and then dividing by the range of the feature. This process ensures that all inputs have a similar scale, which can help improve the performance and convergence speed of learning algorithms. In the context of CSI-based sensing, we can apply max-min scaling to the feature matrix  $F[t]$  as shown in Eq.6, where  $max_F$  and  $min_F$  represent the maximum and minimum values of the feature time series  $F$ , respectively.

$$F[t]' = \frac{F[t] - min_F}{max_F - min_F} \quad (6)$$

*Z-normalization* (or standardization) is another popular feature scaling technique that involves subtracting the feature mean and then dividing it by the standard deviation, resulting in features with a mean of 0 and a standard deviation of 1. This process helps in reducing bias and improving the performance of algorithms sensitive to the variance in data, such as gradient descent-based methods and algorithms assuming features with Gaussian distribution. Z-normalization is applied to a feature matrix  $F[t]$  as shown in Eq.7, where  $\mu_F$  and  $\sigma_F$  represent the mean and standard deviation of  $F$ , respectively.

$$F[t]' = \frac{F[t] - \mu_F}{\sigma_F} \quad (7)$$

### 4.3 Dimensionality Reduction

Dimensionality reduction techniques such as *Principal Component Analysis* (PCA) [11], *Independent Component Analysis* (ICA) [5], and *Uniform Manifold Approximation and Projection* (UMAP) [18] are foundational across various disciplines, primarily for their capacity to distill complex datasets into a more manageable form. For WiFi-based person-centric sensing, these dimensionality reduction techniques offer promising strategies for dealing with high-dimensional data, potentially enhancing model performance and generalization.

PCA compresses datasets by projecting them onto a new coordinate system defined by principal components, which are directions of maximum variance. This approach not only reduces the dimensionality but also manages to eliminate noisy OFDM subcarriers [10], thereby enhancing model performance. ICA distinguishes itself by separating multivariate signals into independent non-Gaussian components. This is especially beneficial in multi-person scenarios within WiFi-based sensing, where it helps identify original signal sources from complex mixtures (blind source separation problem) [30]. UMAP, on the other hand, offers a non-linear approach to dimensionality reduction, effectively maintaining both the local and global structure of high-dimensional data. This method is valuable for exploring complex patterns within data, facilitating insights into intricate relationships that linear techniques like PCA might overlook. Applied to WiFi-based indoor localization, UMAP has demonstrated potential in enhancing model performance [21, 29].

### 4.4 Data Augmentation

To enhance model generalization in through-wall scenarios, we investigate the effectiveness of four random perturbation-based data augmentation techniques: *random magnitude*, *circular rotation* along the time axis, *horizontal flipping* (time axis inversion), and *dropout*. These methods address the challenge of temporal signal variability due to hardware drifts and environmental variations, which can prevent model generalization [12, 20].

*Random magnitude* applies a global scaling factor  $s$  to a feature matrix  $F[t]$ , introducing variability in feature magnitude. This scaling is defined in Eq. 8, where  $s$  and  $x$  denote scale factor and magnitude, respectively. This technique aims to simulate real-world variations in the WiFi signal (e.g., long-term amplitude variations), potentially making the model more robust to such variations.

$$F[t]' = F[t]s, \quad \text{with } s \sim \mathcal{U}(1 - x, 1 + x) \quad (8)$$

*Circular rotation* involves shifting the elements of  $F[t]$  circularly along the time axis, effectively simulating temporal shifts in recorded activities. A specific number of shifts, either to the left or right, introduces temporal diversity to the dataset. As demonstrated in [24], this technique can improve generalization by presenting the model with varied sequences of activities.

*Horizontal flipping* inverts the sequence of events in  $F[t]$ , providing a reversed view of the time series. Inspired by techniques used in the image domain, this approach offers a simple way to increase dataset diversity.

*Dropout*, traditionally used as a regularization technique, is adapted here at the subcarrier and packet levels to mimic destructive interference effects. Unlike typical dropout applications where elements are simply zeroed, our version sets the value of dropped-out elements to the feature mean  $\mu_F$ , maintaining the overall structure of CSI data while introducing randomness. Dropout is implemented as described in Eq. 9, with  $M[t]$  being the binary dropout matrix,  $\neg M[t]$  being its negation and  $\odot$  representing the Hadamard product. We employ subcarrier- and packet-wise dropout [22]. For subcarrier-wise dropout, elements in  $M[t]$  are sampled independently from a Bernoulli distribution with probability  $p$ . For packet-wise dropout, the entries in  $M[t]$  are sampled on a per-column basis.

$$F[t]' = D_\mu(F[t], p) = F[t] \odot M[t] + \neg M[t]\mu_F \quad (9)$$

## 5 Evaluation

The effectiveness of the methods described in Sect. 4 is evaluated through an ablation study, where we train deep learning models for the joint goal of 3D person localization and activity recognition using the discussed techniques. The *EfficientNetV2 small* architecture [26], implemented in *torchvision.models*<sup>8</sup> is chosen for this purpose due to its lightweight design, training speed, and reproducibility. To support the joint objective, we modify the *EfficientNetV2 small* architecture by incorporating an extra head for the 3D regression task. An implementation of this modified architecture is provided (See footnote 1).

### 5.1 Model Training

All models are trained exclusively with data collected on day 1, divided into training and validation subsets at an 8:2 split ratio. A balanced random sampler is employed to mitigate class imbalance effects. The modified *EfficientNetV2 small* architecture, tailored for dual objectives of person localization and activity recognition, is trained with the AdamW optimizer [16] and a cosine annealing learning rate scheduler [15]. For regression and classification tasks, Mean Squared Error (MSE) and Cross-Entropy (CE) losses are combined in the loss  $\mathcal{L} = MSE + CE\alpha$ . The coefficient  $\alpha = 0.4$ , chosen to balance the tasks and prevent overfitting on classification, is determined via a hyperparameter search across  $\alpha \in \{0.1, 0.2, \dots, 1.0\}$ . Further optimization for the learning rate  $l \in \{0.0001, 0.0005, 0.001, 0.0015, 0.002\}$ , batch size  $b \in \{4, 8, 16, 32, 64, 128\}$ , and window size  $w \in \{51, 101, 151, \dots, 351, 401, 451\}$  yields the optimal parameters  $l = 0.001$ ,  $b = 32$ , and  $w = 351$  ( $\sim 3.5$  s at a 100 Hz packet sending rate) which are used for all training runs.

<sup>8</sup> PyTorch *torchvision.models*, <https://pytorch.org>, accessed: 12-03-2024.



Training is conducted in two stages. Initially, we assess the generalization of CSI features, including amplitude, phase, first-order amplitude and phase differences, and PSD. Following this, the feature that shows the best performance is chosen as a baseline. We then apply feature scaling, dimensionality reduction, and data augmentation to this baseline to further enhance model generalization. For each configuration, we conduct three independent, from-scratch training runs spanning 25 epochs, where we select the model checkpoint with the lowest validation loss as the definitive model. Finally, we report the mean and standard deviation for metrics such as Root Mean Squared Error (RMSE), Precision (P), Recall (R), F1-score (F1), and Classification Accuracy (ACC), based on the test data from days 2 and 3.

**Table 2.** Generalization performance of models trained on amplitude ( $\mathcal{A}$ ), phase ( $\mathcal{P}$ ), first-order difference of amplitude ( $\mathcal{A}_\Delta$ ), first-order difference of phase ( $\mathcal{P}_\Delta$ ), and PSD features ( $\mathcal{PSD}$ ). All models are trained on day 1 data using an 8:2 training-validation split and tested on days 2 and 3. The "Day" column indicates the data used: day 1 for validation and days 2 and 3 for testing, with all sequences from each test day used without splitting. Metrics are presented as the mean and standard deviation across three independent training runs.

Model	Day	RMSE [m] ↓	P ↑	R ↑	F1 ↑	ACC ↑
$\mathcal{A}$	1 (val.)	<b>0.364</b> ± 0.00	<b>97.81</b> ± 0.49	<b>99.12</b> ± 0.10	<b>98.46</b> ± 0.23	<b>99.55</b> ± 0.11
$\mathcal{P}$	1 (val.)	0.512 ± 0.02	74.34 ± 8.16	91.66 ± 2.93	81.96 ± 6.24	91.50 ± 2.92
$\mathcal{A}_\Delta$	1 (val.)	0.596 ± 0.02	79.71 ± 5.39	92.50 ± 2.03	85.58 ± 3.98	93.03 ± 1.99
$\mathcal{P}_\Delta$	1 (val.)	0.708 ± 0.01	62.48 ± 1.32	80.18 ± 1.84	70.22 ± 1.16	80.23 ± 1.70
$\mathcal{PSD}$	1 (val.)	<u>0.415</u> ± 0.01	<u>92.81</u> ± 0.76	<u>97.56</u> ± 0.39	<u>95.13</u> ± 0.56	<u>97.97</u> ± 0.47
$\mathcal{A}$	2 (test)	<b>0.587</b> ± 0.02	<b>79.38</b> ± 2.59	<b>82.94</b> ± 3.22	<b>81.12</b> ± 2.89	<b>83.36</b> ± 3.21
$\mathcal{P}$	2 (test)	0.886 ± 0.03	46.33 ± 4.45	48.78 ± 2.50	47.49 ± 3.50	48.97 ± 2.63
$\mathcal{A}_\Delta$	2 (test)	0.689 ± 0.02	63.35 ± 2.79	67.81 ± 2.50	65.51 ± 2.66	68.18 ± 2.53
$\mathcal{P}_\Delta$	2 (test)	0.948 ± 0.02	38.73 ± 3.30	40.46 ± 1.36	39.55 ± 2.32	40.73 ± 1.39
$\mathcal{PSD}$	2 (test)	<u>0.588</u> ± 0.01	<u>76.46</u> ± 0.87	<u>80.11</u> ± 1.47	<u>78.24</u> ± 1.15	<u>80.57</u> ± 1.50
$\mathcal{A}$	3 (test)	<b>0.904</b> ± 0.01	<b>77.52</b> ± 2.87	<b>80.63</b> ± 4.67	<b>79.04</b> ± 3.71	<b>81.16</b> ± 4.57
$\mathcal{P}$	3 (test)	0.951 ± 0.02	64.74 ± 3.56	66.81 ± 6.62	65.72 ± 5.04	67.36 ± 6.55
$\mathcal{A}_\Delta$	3 (test)	0.972 ± 0.03	<u>71.35</u> ± 0.07	<u>76.92</u> ± 0.30	<u>74.03</u> ± 0.14	<u>77.64</u> ± 0.30
$\mathcal{P}_\Delta$	3 (test)	0.962 ± 0.03	64.06 ± 1.71	73.81 ± 2.50	68.58 ± 1.93	74.27 ± 2.44
$\mathcal{PSD}$	3 (test)	<u>0.939</u> ± 0.01	67.97 ± 0.46	71.69 ± 0.37	69.78 ± 0.36	72.21 ± 0.24

## 5.2 Results

*CSI Feature Selection.* Table 2 presents the performance of models utilizing various CSI features: amplitude ( $\mathcal{A}$ ), phase ( $\mathcal{P}$ ), first-order difference in amplitude ( $\mathcal{A}_\Delta$ ), first-order difference in phase ( $\mathcal{P}_\Delta$ ), and PSD ( $\mathcal{PSD}$ ). The models show a decline in performance on days 2 and 3 compared to the first day (validation),

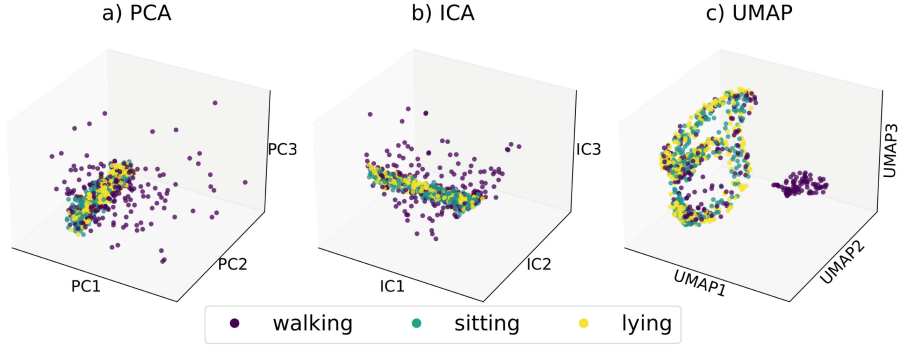
highlighting a significant domain gap that negatively affects generalization. This performance drop aligns with our experimental design, which introduces temporal and dynamic environmental variations on day 2 and adds static environmental variations on day 3.

Among the CSI features evaluated,  $\mathcal{A}$  demonstrates the best generalization capability, achieving the lowest RMSE and highest P, R, F1, and ACC on both days 2 and 3. Compared to  $\mathcal{P}$ ,  $\mathcal{A}$  shows a 50.94% and 5.19% reduction in RMSE and a 41.25% and 17.00% increase in ACC on days 2 and 3, respectively, underscoring its superiority in our through-wall scenario. PSD features rank second in generalization performance, with  $\mathcal{PSD}$  recording the lowest RMSE on both days and the highest P, R, F1, and ACC on day 2, only to be surpassed by  $\mathcal{A}_\Delta$  in these metrics on day 2.  $\mathcal{P}$  exhibits the poorest generalization. Despite the theoretical assumption that first-order difference features are less environmentally dependent [1,6], our results indicate otherwise in through-wall scenarios. Although  $\mathcal{A}_\Delta$  shows promising classification performance on day 2, it and  $\mathcal{P}_\Delta$  are outperformed by models utilizing plain amplitude features  $\mathcal{A}$ . This finding is consistent with LoS scenarios [10], where our day 2 results (without static environmental variations) are directly comparable to the medium and large-scale person-centric sensing experiments conducted. Notably, comparing the feature performance rankings, with  $\mathcal{A}$  at the top and first-order difference phase features  $\mathcal{P}_\Delta$  at the bottom, suggests that feature selection effectiveness in LoS scenarios translates well to through-wall scenarios.

In summary, our evaluation reveals that in through-wall scenarios, opting for features other than amplitude provides no significant advantage.  $\mathcal{A}$  not only responds better to environmental variations but also offers greater computational efficiency than derivative features like first-order differences or PSD. Consequently,  $\mathcal{A}$  is chosen as the baseline for evaluating the effectiveness of feature scaling, dimensionality reduction and data augmentation techniques.

*Feature Scaling.* The results of applying feature scaling methods are detailed in Table 3. Utilizing max-min scaling, denoted as  $\mathcal{A}_{mm}$ , results in consistent improvements in the baseline performance, represented by  $\mathcal{A}$ , across both localization and classification metrics on days 2 and 3. Conversely, z-normalization, indicated by  $\mathcal{A}_z$ , improves localization accuracy on days 2 and 3 but exhibits a slight decrease in classification performance on these days when compared to the baseline. While max-min scaling displays marginally better generalization capabilities than z-normalization, the observed differences are minimal and could be attributed to variability in model training. Therefore, our findings do not conclusively favor either max-min scaling or z-normalization as the superior feature scaling technique.

*Dimensionality Reduction.* The performance of models trained on amplitude features subjected to dimensionality reduction using PCA ( $\mathcal{A}_{PCA\_d}$ ), ICA ( $\mathcal{A}_{ICA\_d}$ ), and UMAP ( $\mathcal{A}_{UMAP\_d}$ ) is detailed in Table 3. The number  $d$  in the model names signifies the reduced dimensionality, which was optimized through



**Fig. 4.** Comparison of dimensionality reduction techniques on day 1 data, showing a) PCA, b) ICA, and c) UMAP projections down to three dimensions (visualizing 0.5% of day 1 samples). We observe that neither PCA nor ICA effectively separates activity clusters. In contrast, UMAP distinguishes most samples associated with the *walking* activity from the combined cluster of *sitting* and *lying* activities.

a hyperparameter search within the range  $d \in \{2, 3, 4, \dots, 52\}$ , utilizing day 1 validation data to determine the best  $d$  values for each method. This search yielded an optimal dimensionality of 42 for PCA, 24 for ICA, and 48 for UMAP. The configuration of additional UMAP parameters is provided in the supplementary material.

Our evaluation of dimensionality reduction techniques shows that none of the applied methods enhance performance metrics beyond the baseline. PCA and UMAP demonstrate comparable performance levels, with PCA having a slight edge on day 2 and UMAP on day 3. In contrast, ICA leads to significant performance degradation, with a reduction in ACC of 21.55 % on day 2 and 33.67 % on day 3 relative to the baseline. Figure 4 illustrates the three-dimensional projections of day 1 data using PCA, ICA, and UMAP, showcasing UMAP’s ability to distinguish *walking* samples from the conjoined clusters of *sitting* and *lying* more effectively than PCA and ICA. Despite this advantage in data visualization, UMAP does not lead to better localization and classification performance compared to PCA on the validation set. UMAP’s emphasis on preserving local structures for visualization might lead to a loss of predictive information, in contrast to PCA’s approach of retaining global variance, which could be more relevant for certain predictive tasks. While some studies have reported improved model performance in WiFi-based person-centric sensing tasks with these methods [9, 21], our results in through-wall scenarios do not support these findings.

*Data Augmentation.* In our effort to enhance model robustness and generalization in through-wall scenarios, we evaluate the impact of various data augmentation techniques on amplitude features. This investigation includes random amplitude perturbations, dropout, circular rotation, and horizontal flipping, with

**Table 3.** Effects on model generalization performance of max-min scaling ( $\mathcal{A}$ ), z-normalization ( $\mathcal{A}_z$ ), PCA ( $\mathcal{A}_{PCA\_42}$ ), ICA ( $\mathcal{A}_{ICA\_24}$ ), UMAP ( $\mathcal{A}_{UMAP\_48}$ ), data augmentation  $\mathcal{A}_{AUG}$ , max-min scaling with data augmentation ( $\mathcal{A}_{mmAUG}$ ) and z-normalization with data augmentation ( $\mathcal{A}_{zAUG}$ ). All models are trained on day 1 data using an 8:2 training-validation split and tested on days 2 and 3. The "Day" column indicates the data used: day 1 for validation and days 2 and 3 for testing, with all sequences from each test day used without splitting. Metrics are presented as the mean and standard deviation across three independent training runs.

Model	Day	RMSE [m] ↓	P ↑	R ↑	F1 ↑	ACC ↑
$\mathcal{A}$ (baseline)	1 (val.)	0.364 ± 0.00	97.81 ± 0.49	<b>99.12</b> ± 0.10	<b>98.46</b> ± 0.23	<b>99.55</b> ± 0.11
$\mathcal{A}_{mm}$	1 (val.)	0.363 ± 0.01	97.41 ± 0.28	98.70 ± 0.20	98.05 ± 0.23	99.26 ± 0.24
$\mathcal{A}_z$	1 (val.)	<u>0.356</u> ± 0.01	<b>98.04</b> ± 0.17	98.71 ± 0.10	98.37 ± 0.06	<u>99.45</u> ± 0.05
$\mathcal{A}_{PCA\_42}$	1 (val.)	0.360 ± 0.01	97.58 ± 0.16	98.60 ± 0.15	98.08 ± 0.06	99.49 ± 0.04
$\mathcal{A}_{ICA\_24}$	1 (val.)	0.770 ± 0.28	64.54 ± 25.4	69.18 ± 25.6	66.73 ± 25.4	69.33 ± 25.4
$\mathcal{A}_{UMAP\_48}$	1 (val.)	0.467 ± 0.01	93.54 ± 1.14	97.78 ± 0.33	95.61 ± 0.76	98.19 ± 0.19
$\mathcal{A}_{AUG}$	1 (val.)	0.347 ± 0.00	<u>97.92</u> ± 0.76	<u>98.93</u> ± 0.21	<u>98.42</u> ± 0.48	99.32 ± 0.20
$\mathcal{A}_{mmAUG}$	1 (val.)	0.350 ± 0.01	97.70 ± 0.36	98.86 ± 0.18	98.27 ± 0.23	99.25 ± 0.11
$\mathcal{A}_{zAUG}$	1 (val.)	<b>0.343</b> ± 0.00	97.91 ± 0.11	98.82 ± 0.06	98.37 ± 0.08	99.33 ± 0.14
$\mathcal{A}$ (baseline)	2 (test)	0.587 ± 0.02	79.38 ± 2.59	82.94 ± 3.22	81.12 ± 2.89	83.36 ± 3.21
$\mathcal{A}_{mm}$	2 (test)	0.542 ± 0.06	80.48 ± 3.45	83.19 ± 3.58	81.81 ± 3.52	83.54 ± 3.65
$\mathcal{A}_z$	2 (test)	0.573 ± 0.02	78.53 ± 0.82	82.77 ± 2.03	80.59 ± 1.34	83.10 ± 2.04
$\mathcal{A}_{PCA\_42}$	2 (test)	0.601 ± 0.05	76.46 ± 1.84	80.72 ± 2.82	78.53 ± 2.30	81.07 ± 2.88
$\mathcal{A}_{ICA\_24}$	2 (test)	0.796 ± 0.16	61.79 ± 12.8	65.13 ± 13.8	63.42 ± 13.3	65.40 ± 14.0
$\mathcal{A}_{UMAP\_48}$	2 (test)	0.720 ± 0.02	72.23 ± 1.44	72.06 ± 2.10	72.14 ± 1.77	72.23 ± 2.11
$\mathcal{A}_{AUG}$	2 (test)	<b>0.500</b> ± 0.02	82.20 ± 1.46	<u>86.23</u> ± 1.37	84.17 ± 1.42	<u>86.52</u> ± 1.38
$\mathcal{A}_{mmAUG}$	2 (test)	<u>0.503</u> ± 0.01	<b>84.22</b> ± 0.19	<b>87.60</b> ± 0.72	<b>85.88</b> ± 0.44	<b>88.00</b> ± 0.75
$\mathcal{A}_{zAUG}$	2 (test)	0.527 ± 0.02	<u>82.80</u> ± 1.91	86.18 ± 2.21	<u>84.45</u> ± 2.05	86.51 ± 2.22
$\mathcal{A}$ (baseline)	3 (test)	0.904 ± 0.01	77.52 ± 2.87	80.63 ± 4.67	79.04 ± 3.71	81.16 ± 4.57
$\mathcal{A}_{mm}$	3 (test)	0.887 ± 0.01	<b>81.91</b> ± 7.59	<b>83.31</b> ± 8.15	<b>82.60</b> ± 7.86	<b>83.89</b> ± 8.07
$\mathcal{A}_z$	3 (test)	0.896 ± 0.01	77.67 ± 2.19	80.00 ± 2.36	78.82 ± 2.24	80.69 ± 2.29
$\mathcal{A}_{PCA\_42}$	3 (test)	0.948 ± 0.02	66.18 ± 6.89	68.32 ± 8.07	67.23 ± 7.46	68.89 ± 8.01
$\mathcal{A}_{ICA\_24}$	3 (test)	1.012 ± 0.04	52.99 ± 15.6	53.46 ± 15.5	53.19 ± 15.5	53.83 ± 15.8
$\mathcal{A}_{UMAP\_48}$	3 (test)	0.913 ± 0.01	74.56 ± 2.36	79.24 ± 1.78	76.83 ± 2.09	79.91 ± 1.72
$\mathcal{A}_{AUG}$	3 (test)	<b>0.871</b> ± 0.00	77.93 ± 4.00	79.68 ± 3.74	78.79 ± 3.84	80.31 ± 3.64
$\mathcal{A}_{mmAUG}$	3 (test)	<u>0.872</u> ± 0.02	79.71 ± 1.97	<u>82.10</u> ± 2.75	80.88 ± 2.35	<u>82.76</u> ± 2.76
$\mathcal{A}_{zAUG}$	3 (test)	<u>0.880</u> ± 0.02	<u>79.94</u> ± 2.05	82.02 ± 1.82	<u>80.97</u> ± 1.93	82.61 ± 1.86

optimal parameters identified through a comprehensive hyperparameter search. For conciseness, we include detailed results in the supplementary material. Our findings reveal that neither random amplitude perturbations nor dropout consistently improve performance over the baseline on days 2 and 3. This lack of improvement is attributed to the temporal stability of our WiFi system, reflected by minimal variation in mean ± standard deviation measurements across days (day 1: 12.90 ± 2.33, day 2: 12.90 ± 2.27, day 3: 12.77 ± 2.55), indicating that

such perturbations diverge from the inherent data distribution and result in degraded test performance.

Contrastingly, random circular rotations and horizontal flipping significantly enhance localization and classification performance. Specifically, circular rotation with a magnitude of  $\pm 12.5\%$  (or  $\pm 43$  samples) ( $\mathcal{A}_{AUG}$ ), as detailed in Table 3, delivers the largest improvement, reducing RMSE by 14.82% on day 2 and 3.65% on day 3. Furthermore, this technique increases ACC by 3.98% on day 2, while rotations of  $\pm 6.25\%$  enhance ACC by 2.24% on day 3, suggesting optimal rotation magnitudes may be dataset-specific. To determine if the benefits of these augmentations are additive, we combine circular rotations at  $\pm 12.5\%$  with horizontal flipping. This combination leads to a 14.31% reduction in RMSE on day 2 and a 2.10% reduction on day 3, alongside a 3.42% increase in ACC on day 2 but a decrease of 1.66% on day 3. Hence, this approach slightly underperforms compared to using circular rotation augmentation alone, highlighting the complex interactions between different augmentation techniques.

Finally, combining circular rotations at  $\pm 12.5\%$  with max-min scaling ( $\mathcal{A}_{mmAUG}$ ) and z-normalization ( $\mathcal{A}_{zAUG}$ ), we observe that max-min scaling yields improved performance. While RMSE on both days remains stable, day 2 experiences the highest F1 and ACC scores at 85.88% and 88.00%, respectively. On day 3, there is a noticeable improvement over the baseline, yet  $\mathcal{A}_{mmAUG}$  does not surpass  $\mathcal{A}_{mm}$  in classification performance, indicating that max-min scaling’s effectiveness may depend on its combination with specific augmentation techniques.

## 6 Limitations and Future Work

Our evaluation highlights the superior generalization of plain amplitude features in through-wall scenarios, consistent with findings from LoS scenarios [10]. However, our scenarios represent only a subset of potential domain variations. Future work should explore a broader range of scenarios, including different transmitter-receiver arrangements, antenna types, and diverse physiological characteristics of multiple participants to validate the robustness and transferability of our findings across various real-world contexts. Another promising area for exploration is the combination of multiple CSI features. For example, combining amplitude and phase features could enhance model performance [13]. Investigating these combinations may yield significant improvements in generalization capabilities. Our analysis, which first selects a baseline feature and then evaluates additional methods based on this baseline, opens avenues for further research into non-selected features. Features such as PSD, paired with complementary methods, could achieve comparable or superior generalization performance. Exploring these alternatives will provide a more comprehensive understanding of feature-method interactions and their impacts on model performance. To further validate our findings and place our work in the context of related studies, conducting generalization experiments using other publicly available WiFi CSI datasets is essential. This critical next step will help ascertain the broader applicability

and robustness of our methods. We also observed that traditional dimensionality reduction techniques like PCA, commonly effective in LoS scenarios, did not enhance performance in through-wall scenarios. This suggests unique characteristics of through-wall propagation that warrant deeper investigation. Future research could focus on developing dimensionality reduction techniques better suited to through-wall scenarios.

In summary, while our evaluation provides a strong foundation, there are numerous opportunities for expanding and validating our findings. By exploring diverse scenarios, combining multiple CSI features, conducting generalization experiments with additional datasets, investigating alternative features and methods, and refining dimensionality reduction techniques, future research can significantly advance the development of robust through-wall person-centric sensing systems.

## 7 Privacy and Ethical Considerations

WiFi-based sensing technology introduces significant privacy and ethical considerations due to its ability to monitor individuals without explicit consent. While this technology can be highly beneficial, especially in privacy-sensitive applications like assisted living where it can serve as a visual-privacy preserving alternative to optical modalities [2], its use must be carefully managed to prevent misuse. The primary concern is the potential for unauthorized monitoring, as WiFi signals can penetrate walls, raising the risk of inadvertent surveillance [9]. Ethical deployment of this technology requires obtaining informed consent from those being monitored and ensuring transparency about data collection and usage. Robust privacy-preserving mechanisms, such as data anonymization, stringent access controls, and data security measures, should be implemented to mitigate privacy risks. Additionally, the use of WiFi-based sensing should be restricted to environments where there are clear benefits and explicit user consent has been obtained. By addressing these privacy and ethical issues, WiFi-based sensing can be leveraged responsibly and effectively, ensuring that the technology respects individuals' privacy and upholds ethical standards.

## 8 Conclusion

In this work, we conducted a comprehensive evaluation of methods to enhance model generalization in through-wall person-centric sensing scenarios, focusing on CSI feature selection, feature scaling, dimensionality reduction, and data augmentation techniques. Our approach involved collecting a dataset over three days, including CSI and 3D trajectory data across three activity classes under static, dynamic, and temporal environmental variations, which is made publicly available to stimulate further research on generalizable person-centric sensing in through-wall scenarios (See Footnote 1). Our findings reveal that, in through-wall scenarios, models leveraging plain amplitude features consistently demonstrate superior generalization across environmental variations. This superiority

is further enhanced through the application of feature scaling methods such as max-min scaling and z-normalization, as well as data augmentation techniques such as random circular rotation and horizontal flipping. Contrary to expectations, our exploration into dimensionality reduction methods, including PCA, ICA, and UMAP, did not yield improvements in model generalization, suggesting that findings on LoS scenarios are not directly transferable to through-wall scenarios.

## References

1. Ali, K., Alloulah, M., Kawsar, F., Liu, A.X.: On goodness of WiFi based monitoring of sleep vital signs in the wild. *IEEE Trans. Mob. Comput.* **22**(1), 341–355 (2023). <https://doi.org/10.1109/TMC.2021.3077533>
2. Arning, K., Ziefle, M.: “Get that camera out of my house!” Conjoint measurement of preferences for video-based healthcare monitoring systems in private and public places. In: Geissbühler, A., Demongeot, J., Mokhtari, M., Abdulrazak, B., Aloulou, H. (eds.) *Inclusive Smart Cities and e-Health*, pp. 152–164. Springer, Cham (2015)
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans. Robot.* **37**(6), 1874–1890 (2021)
4. Chen, C., Zhou, G., Lin, Y.: Cross-domain WiFi sensing with channel state information: a survey. *ACM Comput. Surv.* **55**(11), 1–37 (2023)
5. Comon, P.: Independent component analysis, a new concept? *Signal Process.* **36**(3), 287–314 (1994)
6. Ding, E., Li, X., Zhao, T., Zhang, L., Hu, Y., et al.: A robust passive intrusion detection system with commodity WiFi devices. *J. Sens.* **2018** (2018)
7. Fu, B., Damer, N., Kirchbuchner, F., Kuijper, A.: Sensing technology for human activity recognition: a comprehensive survey. *IEEE Access* **8**, 83791–83820 (2020). <https://doi.org/10.1109/ACCESS.2020.2991891>
8. Gao, K., Wang, H., Lv, H., Liu, W.: Toward 5G NR high-precision indoor positioning via channel frequency response: a new paradigm and dataset generation method. *IEEE J. Sel. Areas Commun.* **40**(7), 2233–2247 (2022)
9. Hernandez, S.M., Bulut, E.: Adversarial occupancy monitoring using one-sided through-wall WiFi sensing. In: *ICC 2021 - IEEE International Conference on Communications*, pp. 1–6 (2021). <https://doi.org/10.1109/ICC42927.2021.9500267>
10. Hernandez, S.M., Bulut, E.: WiFi sensing on the edge: signal processing techniques and challenges for real-world systems. *IEEE Commun. Surv. Tutor.* **25**(1), 46–76 (2023). <https://doi.org/10.1109/COMST.2022.3209144>
11. Jolliffe, I.T.: *Principal component analysis for special types of data*. Springer (2002)
12. Lee, H., Ahn, C.R., Choi, N.: Toward single occupant activity recognition for long-term periods via channel state information. *IEEE Internet Things J.* **1** (2023)
13. Li, B., Cui, W., Wang, W., Zhang, L., Chen, Z., Wu, M.: Two-stream convolution augmented transformer for human activity recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 286–293 (2021)
14. Liu, Y., et al.: UniFi: a unified framework for generalizable gesture recognition with Wi-Fi signals using consistency-guided multi-view networks. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **7**(4) (2024)
15. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. arXiv preprint [arXiv:1608.03983](https://arxiv.org/abs/1608.03983) (2016)

16. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
17. Ma, Y., Zhou, G., Wang, S.: WiFi sensing with channel state information: a survey. *ACM Comput. Surv.* **52** (2019). <https://doi.org/10.1145/3310194>
18. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426) (2018)
19. Parameswaran, A.T., Husain, M.I., Upadhyaya, S., et al.: Is RSSI a reliable parameter in sensor localization algorithms: an experimental study. In: *Field Failure Data Analysis Workshop (F2DA09)*, Niagara Falls, NY, USA, vol. 5. IEEE (2009)
20. Serbetci, O.G., Lee, J.H., Burghal, D., Molisch, A.F.: Simple and effective augmentation methods for CSI based indoor localization (2023)
21. Stahlke, M., Yammine, G., Feigl, T., Eskofier, B.M., Mutschler, C.: Indoor localization with robust global channel charting: a time-distance-based approach. *IEEE Trans. Mach. Learn. Commun. Netw.* **1**, 3–17 (2023). <https://doi.org/10.1109/TMLCN.2023.3256964>
22. Strohmayer, J., Kampel, M.: A compact tri-modal camera unit for RGBDT vision. In: *2022 The 5th International Conference on Machine Vision and Applications (ICMVA 2022)*, New York, NY, USA, pp. 34–42 (2022). <https://doi.org/10.1145/3523111.3523116>
23. Strohmayer, J., Kampel, M.: WiFi CSI-based long-range through-wall human activity recognition with the ESP32. In: *Computer Vision Systems*, pp. 41–50. Springer, Cham (2023)
24. Strohmayer, J., Kampel, M.: Data augmentation techniques for cross-domain WiFi CSI-based human activity recognition. arXiv preprint [arXiv:2401.00964](https://arxiv.org/abs/2401.00964) (2024)
25. Strohmayer, J., Kampel, M.: WiFi CSI-based long-range person localization using directional antennas. In: *The Second Tiny Papers Track at ICLR 2024* (2024). <https://openreview.net/forum?id=AOJFcEh5Eb>
26. Tan, M., Le, Q.V.: EfficientNetV2: smaller models and faster training (2021)
27. Wang, D., Yang, J., Cui, W., Xie, L., Sun, S.: AirFi: empowering WiFi-based passive human gesture recognition to unseen environment via domain generalization. *IEEE Trans. Mob. Comput.* **23**(2), 1156–1168 (2024). <https://doi.org/10.1109/TMC.2022.3230665>
28. Wang, X., Wang, Y., Wang, D.: A real-time CSI-based passive intrusion detection method. In: *2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 1091–1098 (2020). <https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00163>
29. Xu, Z., Huang, B., Jia, B., Li, W., Lu, H.: A boundary aware WiFi localization scheme based on UMAP and KNN. *IEEE Commun. Lett.* **26**(8), 1789–1793 (2022). <https://doi.org/10.1109/LCOMM.2022.3179447>
30. Zeng, Y., Wu, D., Xiong, J., Liu, J., Liu, Z., Zhang, D.: MultiSense: enabling multi-person respiration sensing with commodity WiFi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **4**(3), 1–29 (2020)
31. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization (2018)
32. Zhang, H., et al.: Understanding the mechanism of through-wall wireless sensing: a model-based perspective. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **6**(4) (2023). <https://doi.org/10.1145/3569494>





# Towards Open-Set Egocentric Action Recognition with Uncertainty Estimation

Yishan Zou<sup>1,2</sup>, Christopher Nugent<sup>2</sup>, Matthew Burns<sup>2</sup>, Xiaoming Xi<sup>1</sup>,  
and Meng Liu<sup>1</sup>(✉)

<sup>1</sup> Shandong Jianzhu University, Jinan, Shandong, China  
fyzq10@126.com, mengliu.sdu@gmail.com

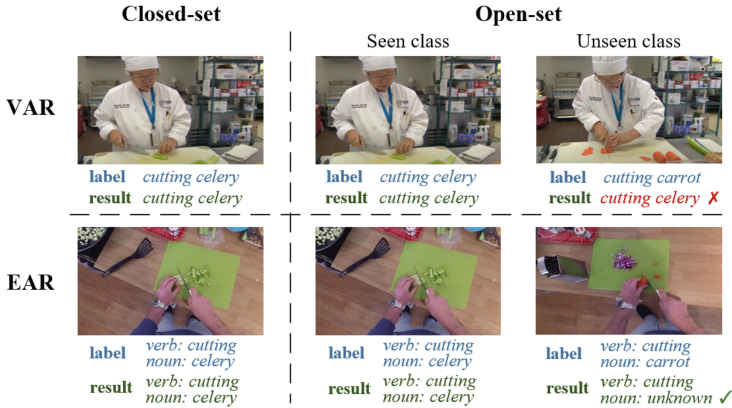
<sup>2</sup> Ulster University, Belfast BT15 1AP, UK  
{zou-y, cd.nugent, m.burns2}@ulster.ac.uk, fyzq10@126.com

**Abstract.** Egocentric Action Recognition (EAR) has gained significant attention due to its widespread applicability in lifestyle analysis, medical monitoring, and industrial robotics, among other real-world scenarios. However, existing EAR methods are built on the closed-set assumption, making it challenging to handle unknown actions inevitably present in open-world scenarios and struggling to meet the dual requirements of accuracy and reliability while providing decisions. To address the Open-set EAR problem, this paper presents a Open-set Egocentric Action Recognition (OpenEAR) framework, advancing beyond traditional egocentric action recognition methods. OpenEAR distinguishes itself by adeptly handling unknown actions in open-world scenarios, a notable limitation in conventional EAR models. Utilizing large-scale pre-trained models and refined architecture, OpenEAR excels in semantic extraction from egocentric videos. Its unique incorporation of Evidential Deep Learning (EDL) allows for uncertainty estimation, enhancing prediction reliability. This novel approach not only recognizes known actions and objects but also quantifies prediction confidence, effectively managing unknown elements. Demonstrated superior performance on EPIC-KITCHENS-55 and EGTEA Gaze+ datasets underlines OpenEAR's robustness and practicality, marking a significant leap from existing methods. The OpenEAR framework is available at <https://github.com/zou-y23/OpenEAR>.

**Keywords:** Egocentric action recognition · Open-set recognition · Uncertainty estimation

## 1 Introduction

The emergence of egocentric Video Action Recognition (VAR) represents a significant turning point in the field, driven by the widespread use of wearable technologies such as smart glasses. These devices have fundamentally altered data collection methods, offering a first-person perspective that is especially relevant in areas like augmented reality and robotics [1]. This perspective enables



**Fig. 1.** In closed-set conditions, the system accurately identifies the action and object as "cutting celery". In open-set conditions, the system correctly labels seen classes but may misidentify unseen classes, such as mistaking a "cutting carrot" action for "cutting celery". Open-set EAR approaches can correctly identify actions while acknowledging unknown objects, improving the recognition accuracy.

a deeper understanding of the user's interactions, enhancing user experience across various applications, from personal assistance to healthcare monitoring and interactive gaming.

However, egocentric VAR faces unique challenges, particularly in Open-Set Recognition (OSR). In OSR, systems need to identify both familiar and novel action categories [2,3], labeling the latter as "unknown", as illustrated in Fig. 1. This necessity arises from the limitations of closed-set environments, where classifiers are trained and tested on predetermined categories. The real world, in contrast, constantly presents new actions, demanding adaptability from recognition systems. Current research in egocentric VAR, largely focused on closed-set scenarios, often fails to address the complexities of open-set environments effectively. This shortcoming underscores the need for systems capable of recognizing a broad spectrum of actions with high accuracy and reliability, especially when encountering novel, unseen actions.

The application of methodologies from exocentric VAR [4,5] to egocentric videos reveals distinct limitations. While exocentric VAR has advanced through significant research and the development of sophisticated models and datasets, these do not readily apply to egocentric VAR. The challenges stem from differences in data characteristics and the contextual interpretation of actions between the two perspectives. Egocentric videos, often unprocessed and subject to motion blur, require simultaneous action and object recognition, such as predicting the verb "cutting" and the noun "celery" in Fig. 1, a demand less critical in exocentric VAR.

In response to these challenges, this paper introduces OpenEAR, a novel framework tailored for open-set egocentric VAR. OpenEAR leverages a large-

scale pre-trained model, undergoing specific architectural modifications to align with the complex scenarios of egocentric videos. These adjustments ensure that OpenEAR can effectively handle the diverse, uncurated content typical of first-person footage. Additionally, OpenEAR integrates the EDL approach [6], employing deep neural networks to predict Dirichlet distributions of class probabilities. This methodology is particularly adept at managing the uncertainty associated with unknown actions, a critical feature for open-set environments. By incorporating EDL, OpenEAR enhances its ability to recognize unfamiliar actions, addressing a key challenge in open-set egocentric VAR and advancing the field towards more adaptable and robust action recognition systems. Experiments on two public datasets demonstrate the effectiveness of our proposed method.

The main contributions of this paper can be summarized as follows:

- We present OpenEAR, an innovative framework designed for action recognition within egocentric video streams under open-set conditions.
- OpenEAR distinguishes itself from conventional approaches by its dual capability to discern recognized actions and objects while concurrently evaluating the certainty of these identifications.
- We have rigorously evaluated OpenEAR, substantiating its efficacy. Meanwhile, we release our code to foster further research and development in the domain of open-set egocentric video action recognition.

## 2 Related Work

### 2.1 Egocentric Action Recognition

Egocentric action recognition is a task aimed at effectively understanding videos captured from a first-person perspective. The key objective is to identify the movements of individuals in the video and their interactions with other objects in the environment. In recent years, to advance research in this field, several egocentric video datasets have emerged [11, 16, 17]. These datasets refine actions into combinations of verbs and nouns, breaking down actions such as "cutting potatoes" into the verb "cut" and the noun "potatoes".

In current literature, an increasing number of studies point out that objects present in videos, especially those relevant to the task, play a crucial role in action recognition [18–20]. Also, other motion cues, such as eye, hand, and head movements [21, 22], are also deemed essential for accurate behavior recognition. Although object-driven approaches currently lead in performance, motion-driven methods may contribute additional robustness to models for EAR. Therefore, hybrid approaches that integrate both object and motion information have gained growing attention in recent years.

In hybrid-driven deep egocentric video analysis methods, the two-stream network [23] has emerged as a popular model. Initially developed for handling exocentric vision, it has been adapted for egocentric videos through refinements. In this regard, the two-stream network proposed in [24] serves as a representative

example. One branch employs a self-attention-based graph convolutional network to capture spatial and short-term temporal information, while the other branch utilizes a bidirectional recurrent neural network for long-term temporal information extraction. Subsequently, in [25], a two-stream network was employed to generate a hierarchical volumetric representation of the 3D environment, enabling the recognition of actions through latent positional and contextual cues. As the natural extension of the two-stream architecture, the development of multi-stream architectures has increasingly become a research focus [26, 27], incorporating additional branches and diverse input modalities. Moreover, some studies have introduced elements such as sound modality [28], multi-task learning [29], and data sampling [30] to enhance the model’s performance and comprehensive understanding.

In the field of self-centric action recognition, while single/hybrid-driven models have provided powerful tools for addressing the task of egocentric video action recognition, there are still some challenges. Firstly, how to effectively utilize both object and motion information within a model, and combining object classification with motion classification to enhance overall action recognition accuracy, is a complex and challenging problem, especially when dealing with dynamic real-world scenarios. Secondly, the complexity of two-stream networks may lead to high computational costs for model training and inference, posing challenges for real-time applicability in practice. In contrast, the method proposed in this paper addresses these concerns by constructing a lightweight action recognition network and introducing uncertainty estimation based on EDL. This approach is more suitable for scenarios with limited computational resources and unknown actions in real-world applications.

## 2.2 Open-set Recognition

The OSR problem, initially gaining prominence in face recognition [31], was formally conceptualized by [2], who introduced a binary Support Vector Machine (SVM) to distinguish unknown classes. This approach, utilizing additional hyperplanes for new classes, laid the groundwork for OSR.

With the advent of deep learning, significant strides have been made in OSR using Deep Neural Networks (DNNs). Bendale et al. [32] proposed OpenMax to mitigate the open-space risk inherent in DNN models, addressing the limitations of softmax. Building on this, Ge et al. [33] introduced G-OpenMax, adopting a generative approach to create unknown samples for DNN training. Generative Adversarial Networks (GANs) and Variational AutoEncoders (VAEs) have also been utilized to generate and assess unknown class samples in OSR [34–38]. However, these methods, often focusing on areas like anomaly detection [40], generalized zero-shot learning [41], and open-world learning [42–44], have limited direct applicability to action recognition. Therefore, this paper concentrates on Open-Set Action Recognition (OSAR).

OSAR is more challenging compared to general OSR, with a limited body of work addressing it. Shu et al. [45] introduced the Open Deep Network (ODN) for gradually incorporating new classes into action recognition. Bayesian techniques

for identifying unfamiliar actions have been explored [46–48], and Busto et al. [49] proposed an open-set domain adaptation method.

Distinguishing our work, we introduce EDL into OSAR, focusing on uncertainty calibration and handling temporal discrepancies in video data. This novel approach enriches the field of OSAR, offering a more comprehensive solution for its inherent challenges.

### 2.3 Evidential Deep Learning

In action recognition, distinguishing between known and unknown samples effectively hinges on developing an Out-Of-Distribution (OOD) scoring function. Recent studies [6, 46, 50–52] have highlighted the potential of using uncertainty predicted by DNNs as a scoring function to identify OOD samples, based on the premise that OOD samples should exhibit higher uncertainty during inference.

Bayesian Neural Networks (BNNs) have been applied in various computer vision tasks to model both aleatoric and epistemic uncertainties [50, 53, 54]. However, BNNs encounter challenges such as complex posterior inference, the need for appropriate weight priors, and high computational costs for uncertainty estimation [55].

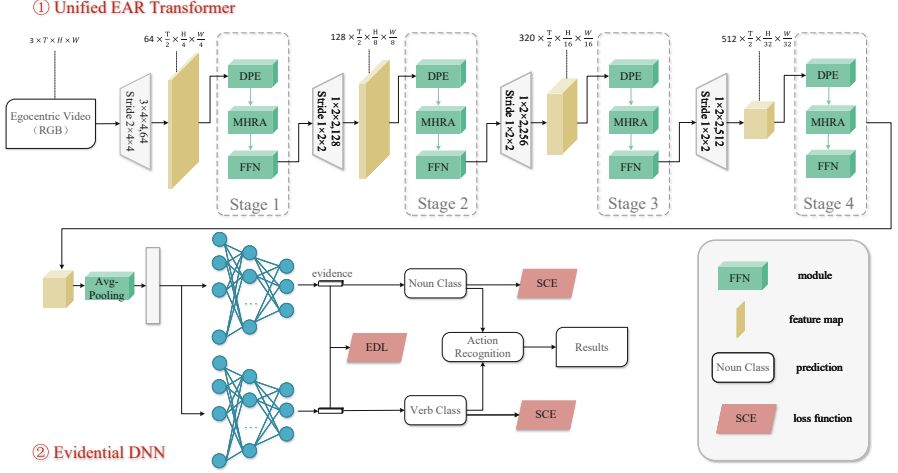
As an alternative, EDL integrates evidence theory with deep neural networks, showing promising results in classification [8] and regression [55] tasks. This paper is a pioneering effort to apply evidence learning to large-scale, uncertainty-aware egocentric action recognition. By adopting EDL, we enhance the assessment of predicted uncertainties, thereby improving the recognition of unknown samples in open-set egocentric video action recognition scenarios.

## 3 Method

The structure and functionality of the OpenEAR model are depicted in Fig. 2. Upon input of a video clip, the model commences with the EAR transformer blocks, which are responsible for feature extraction. These extracted features are subsequently funneled into a bifurcated classifier designed to separately process nouns and verbs. Concurrently, an evidential deep neural network operates in tandem with the classifier to ascertain the definitive action categories, incorporating a measure of uncertainty into the determination. The subsequent subsections of the paper delve into a more granular exposition of each constituent element of the model.

### 3.1 Unified EAR Transformer

In the domain of egocentric video analysis, the temporal consistency of user behaviors presents a unique opportunity for action recognition. Recognizing the pattern that backgrounds usually remain static, user movements are primarily motion-based, and the objects of interaction do not change frequently, we leverage these consistencies to enhance our model. Drawing from the principles outlined in [7], our model incorporates a unified EAR transformer. This transformer



**Fig. 2.** Architecture of the OpenEAR Framework.

is adept at reducing redundancy across video frames and capturing the temporal relationships inherent in the data. Similar to [7], our unified EAR Transformer is comprised of three integral components: Dynamic Position Embedding (DPE), the Multi-Head Relation Aggregator (MHRA), and the Feed-Forward Network (FFN). When an egocentric video, denoted as  $\mathbf{X}_{ev} \in \mathbb{R}^{C \times T \times H \times W}$ , is fed into the system, the model’s learning algorithm unfolds as a structured process,

$$\begin{cases} \mathbf{X} = DPE(\mathbf{X}_{ev}) + \mathbf{X}_{ev}, \\ \mathbf{Y} = MHRA(Norm(\mathbf{X})) + \mathbf{X}, \\ \mathbf{Z} = FFN(Norm(\mathbf{Y})) + \mathbf{Y}, \end{cases} \quad (1)$$

where the DPE is applied to the input egocentric video  $\mathbf{X}_{ev}$  via a 3D Depth Wise Convolution (DWConv), which is characterized by zero padding to maintain the spatial dimensions. The DWConv operation serves to encode the position information dynamically, taking into account the temporal dimension that is critical in understanding the sequence of frames in egocentric videos.

The MHRA executes token relation learning through a multi-head fusion strategy, which can be formalized as

$$\begin{cases} R_n(\mathbf{X}) = \mathbf{A}_n^* V_n(\mathbf{X}^T), \\ MHRA(\mathbf{X}) = Concat(R_1(\mathbf{X}); R_2(\mathbf{X}); \dots; R_N(\mathbf{X}))\mathbf{U}. \end{cases} \quad (2)$$

Given the input tensor  $\mathbf{X} \in \mathbb{R}^{C \times T \times H \times W}$ , it is first reshaped to a sequence of tokens  $\mathbf{X} \in \mathbb{R}^{C \times L}$ , where  $L = T \times H \times W$ .  $R_n(\cdot)$  signifies the function of the Relation Aggregator (RA) in the  $n$ -th head. The term  $\mathbf{U} \in \mathbb{R}^{C \times C}$  represents a learnable matrix that integrates the outputs of  $N$  different heads. Each head, within the RA, is tasked with encoding the context of tokens and learning

token affinities. The tokens undergo a linear transformation to become context  $V_n(\mathbf{X}) \in \mathbb{R}^{L \times \frac{C}{N}}$ , allowing the RA to summarize the context under the guidance of token affinity  $\mathbf{A}_n^* \in \mathbb{R}^{L \times L}$  and  $* \in \{local, global\}$ . A pivotal aspect of the RA is the learning mechanism for the token affinity  $\mathbf{A}_n^*$ , which is essential for understanding the relationships between tokens within video sequences.

Within the architecture of the MHRA, the token affinity learning is stratified across the network’s depth, leveraging both local and global contextual cues. In the shallow (early) layers of the network, the token affinity is approached as a learnable parameter matrix, which operates within a local 3D neighborhood. Here, for a given anchor token  $\mathbf{X}_i$ , RA learns the local spatiotemporal affinity between this token and other tokens within a small, defined tube  $\Omega_i^{t \times h \times w}$ , as denoted by

$$\mathbf{A}_n^{local}(\mathbf{X}_i, \mathbf{X}_j) = a_n^{i-j}, \quad j \in \Omega_i^{t \times h \times w}, \quad (3)$$

where  $a_n \in \mathbb{R}^{t \times h \times w}$  represents the set of learnable parameters, and  $\mathbf{X}_j$  refers to any neighboring token within  $\Omega_i^{t \times h \times w}$  (more details are referenced in [7]). The term  $(i - j)$  indicates the relative index of the token, which is utilized to ascertain the weight for aggregation.

Conversely, in the deeper layers of the network, token affinity learning expands to a global scope. Instead of a local neighborhood, the network examines content similarity across all tokens within the entire video sequence:

$$\mathbf{A}_n^{global}(\mathbf{X}_i, \mathbf{X}_j) = \frac{e^{Q_n(\mathbf{X}_i)^T B_n(\mathbf{X}_j)}}{\sum_{j' \in \Omega_{T \times H \times W}} e^{Q_n(\mathbf{X}_i)^T B_n(\mathbf{X}_{j'})}}. \quad (4)$$

In this equation,  $\mathbf{X}_j$  can be any token from the global 3D space encompassing the entire temporal and spatial dimensions  $T \times H \times W$ . The functions  $Q_n(\cdot)$  and  $B_n(\cdot)$  are linear transformations that project the tokens into spaces where their content-based similarity can be computed. This global view allows the network to understand the broader context and relationships across the entire video, which is critical for recognizing actions in egocentric videos where the scene and user interactions can change dramatically over time.

### 3.2 Egocentric Action Classifier

In analyzing egocentric videos, our objective is to classify the video  $\mathbf{X}_{ev}$  into a set of verb classes, totaling  $M$ , and noun classes, numbering  $N$ . The action depicted in the video is deduced from the conjunction of these verb and noun classifications. As illustrated in Fig. 2, our approach employs dual classifiers to predict the verb and noun categories independently:

$$\mathbf{L}^v = \text{CLS}^v(\text{AvgPooling}(\mathbf{Z})), \quad (5)$$

$$\mathbf{L}^n = \text{CLS}^n(\text{AvgPooling}(\mathbf{Z})), \quad (6)$$

where  $\mathbf{Z} \in \mathbb{R}^{C' \times T' \times H' \times W'}$  represents the output features from the unified EAR transformer. These features are initially pooled using average pooling to transform the dimensions to  $\mathbb{R}^D$ , upon which two separate classifiers, one for verbs

(CLS<sup>v</sup>) and one for nouns (CLS<sup>n</sup>), act to map these pooled features to their respective verb and noun classes.  $\mathbf{L}^v$  and  $\mathbf{L}^n$  are the logits predicted by the verb and noun classifiers respectively and the classification probabilities for verbs and nouns are finally denoted by  $\mathbf{P}^v = \text{softmax}(\mathbf{L}^v)$  and  $\mathbf{P}^n = \text{softmax}(\mathbf{L}^n)$ , respectively.

To refine the classification predictions, we employ a modified version of the soft-target cross-entropy loss, defined as follows:

$$\mathcal{L}_{\text{SCE}} = \sum_{j=1}^K -t_j \log \frac{\exp(\mathbf{L}_j^*)}{\sum_{j=1}^K \exp(\mathbf{L}_j^*)}, \quad (7)$$

where  $K \in \{M, N\}$  and  $* \in \{v, n\}$ . In this loss function,  $t_j$  is the binary indicator within the one-hot encoded vector corresponding to the action label  $y$ . The indicator  $t_j$  assumes a value of 1 exclusively when the true action label  $y$  matches the class  $j$ . This loss function serves to guide the network toward minimizing the discrepancy between the predicted and true labels, thereby enhancing the precision of action recognition in egocentric video classification tasks.

### 3.3 Evidential Action Classification

In the context of  $K$ -way uncertainty-aware classification pertinent to our task, we adopt the principles of EDL as detailed in [8] are employed after both verb and noun classifiers respectively. This approach effectively quantifies classification uncertainty, which is particularly crucial in applications where confidence in the prediction is as important as the prediction itself. We posit a Dirichlet distribution  $\text{Dir}(\mathbf{P}|\boldsymbol{\alpha})$  over the categorical probabilities  $\mathbf{P} \in \mathbb{R}^K$ , with  $\boldsymbol{\alpha} \in \mathbb{R}^K$  representing the distribution's concentration parameters, or the "Dirichlet strength". The learning process involves minimizing the negative log-likelihood loss function:

$$\mathcal{L}_{\text{EDL}}(\boldsymbol{\alpha}) = \sum_{j=1}^K t_j (\log(\mathcal{S}) - \log(\alpha_j)), \quad (8)$$

where  $\mathcal{S} = \sum_j \alpha_j$  denotes the sum of the Dirichlet concentration parameters across the  $K$  classes, representing the total strength of evidence.

In the testing phase, for a given video  $\mathbf{X}_{ev}$ , the verb/noun classification branch, utilizing the unified EAR transformer, produces a vector of non-negative evidence  $\mathbf{e} \in \mathbb{R}_+^K$ . This evidence vector not only provides the logits for the outputted probability but also aligns with the framework of subjective logic and evidence theory. The expected value of the classification probability is then calculated as  $\mathbb{E}[\mathbf{P}] = \boldsymbol{\alpha}/\mathcal{S}$ , where  $\boldsymbol{\alpha} = \mathbf{e} + 1$ , adhering to the tenets of evidence theory and subjective logic [9, 10]. The measure of classification uncertainty is estimated by the equation  $\mathbf{u} = K/\mathcal{S}$ , which provides a quantitative assessment of the confidence in the classification results, with higher uncertainty values indicating lower confidence and vice versa. In our experiments, the threshold of uncertainty is set to 0.1. This means that when  $\mathbf{u} \geq 0.1$ , the model considers



it indicative of uncertainty regarding the prediction result, requiring further processing. Conversely, when  $\mathbf{u} < 0.1$ , it reflects confidence in the correctness of the prediction result.

## 4 Experiments

### 4.1 Datasets

We utilize two large-scale egocentric datasets: EPIC-KITCHENS-55 [11] and EGTEA Gaze+ [12], to benchmark the proposed approach.

We divided each original training set into new training and validation subsets. Given that our method transcends data modality limitations, we employ RGB videos as the default medium for training and testing. We categorize the datasets into "seen" and "unseen" subsets. For the "seen" subsets, we further segregated them into training, validation, and testing sets in an 8:1:1 ratio. Concurrently, for the "unseen" subsets, we further segmented them based on the known status of the action category, yielding three subsets: "action unseen", "object unseen", and "both action and object unseen".

To assess the OOD performance and uncertainty estimation capabilities of our OpenEAR, we randomly selected a specified number of samples from the unknown segments to substitute a defined proportion (25%, 50%) of the samples in the test sets, thereby constructing new test sets inclusive of unknown classes.

### 4.2 Baselines

In our experimental evaluation, we benchmarked the performance of the proposed OpenEAR model against two contemporary approaches: Hierarchical Temporal Transformer (HTT) and Deep Evidential Action Recognition (DEAR).

The HTT model, as described in [13], is an end-to-end framework that emphasizes the extraction and utilization of temporal information for action recognition in egocentric videos. The design of HTT is streamlined to enable end-to-end training, allowing the model to leverage the temporal dimension of video data and facilitate action recognition through a single feedforward pass efficiently.

On the other hand, DEAR, as introduced in [4], incorporates principles of evidential learning and is tailored for large-scale video action recognition challenges. It is particularly adept at managing "unknown" instances, which are prevalent in open-set action recognition tasks. DEAR frames the recognition task within the context of uncertainty estimation, making use of EDL to provide a robust framework for recognizing actions and estimating the model's confidence in its predictions.

Both HTT and DEAR serve as relevant comparative methods for our study. HTT is a unified framework to achieve 3D hand pose estimation and action recognition simultaneously, which provides a direct comparison for temporal feature extraction and action recognition capabilities. DEAR is an open-set action recognition method performances well in general open-set exocentric videos rather

than focusing on addressing fine-grained actions and emphasizing the interaction between actions and objects. It offers a comparison of our model’s ability to handle uncertainty and recognize "unknown" actions within open-sets as well as more fine-grained recognition tasks. By conducting experiments against these benchmark methods, we aim to demonstrate the efficacy of OpenEAR in both recognizing a wide array of actions from egocentric videos and accurately quantifying the uncertainty associated with its predictions.

### 4.3 Implementation Details

The OpenEAR framework is built upon the Uniformer model [7] as its core, chosen for its excellence in semantic extraction and recognition. To suit the dual task of recognizing verbs and nouns in egocentric videos, we modified the architecture to include two separate branches for category prediction, each initialized with pre-trained parameters from large datasets. An EDL network is integrated into each branch for uncertainty estimation. Implemented using the PyTorch framework, the model processes videos segmented into 8 frames, resized to  $224 \times 224$  pixels. Training parameters are set to 100 epochs, with a batch size of 8 and a learning rate of  $10^{-4}$ , balancing training efficiency with the accuracy and generalizability of the model’s predictions.

**Table 1.** Comparison with state-of-the-art methods on EPIC-KITCHENS-55.

Method	Action Acc		Object Acc		Union Acc	
	Top1	Top5	Top1	Top5	Top1	Top5
HTT	0.6015	0.9220	0.5024	0.6814	0.3927	0.6544
DEAR	0.6506	0.9602	0.5524	0.7930	0.4200	0.7700
<b>OpenEAR</b>	<b>0.7074</b>	<b>0.9615</b>	<b>0.6603</b>	<b>0.8479</b>	<b>0.5080</b>	<b>0.8260</b>

**Table 2.** Comparison with state-of-the-art methods on EGTEA Gaze+.

Method	Action Acc		Object Acc		Union Acc	
	Top1	Top5	Top1	Top5	Top1	Top5
HTT	0.6583	0.9352	0.5811	0.7829	0.4733	0.7557
DEAR	0.6771	0.9322	0.6429	0.8591	0.4995	0.8309
<b>OpenEAR</b>	<b>0.7789</b>	<b>0.9847</b>	<b>0.7661</b>	<b>0.9362</b>	<b>0.6543</b>	<b>0.9298</b>

#### 4.4 Performance Comparison

In our study, we conducted a comprehensive comparison of the OpenEAR method against baseline models using two prominent datasets: EPIC-KITCHENS-55 and EGTEA Gaze+. The performance results are systematically presented in Tables 1 and 2. Our evaluation focused on both action accuracy and object accuracy, assessing them under Top1 and Top5 metrics. Additionally, we introduced a "union accuracy" metric, which evaluates the simultaneous correctness of both action and object predictions.

It is evident from this illustration that the OpenEAR method substantially surpasses the baseline models in several key aspects: action accuracy, object accuracy, and the combined union accuracy for actions and objects. Specifically, for action accuracy, our approach surpasses the best baselines by 5.68% on Top1 for EPIC-KITCHENS-55 and by 10.18% on EGTEA Gaze+. For object accuracy, our approach surpasses the best baselines by 10.79% on Top1 for EPIC-KITCHENS-55 and by 12.32% on EGTEA Gaze+. For union accuracy, our approach surpasses the best baselines on Top1 for EPIC-KITCHENS-55 by 8.8%, and by 15.48% on EGTEA Gaze+.

This superiority is noticeable across both datasets, indicating the robustness and effectiveness of OpenEAR in handling the complexities of egocentric video action recognition, particularly in diverse and dynamic real-world environments like kitchens and everyday activities captured in the EGTEA Gaze+ dataset.

#### 4.5 Ablation Study

To better understand the effectiveness of different modules, we conducted ablation study experiments with the following settings: 1) **Ablate backbone:** We replaced the backbone network with other networks which are Video Swin-Transformer [14] and TimeSformer [15]; 2) **Optimise the back:** We froze the parameters of the Unified EAR Transformer and fine-tuned the Evidential DNN module.





As shown in Table 3, each component contributes to our model to a certain extent, emphasizing the necessity of incorporating these mechanisms. It is noteworthy that there is a significant drop in effectiveness at Top1 and Top5 when we freeze the Unified EAR Transformer. Therefore, the Unified EAR Transformer leads to more significant improvements in our model's performance compared to the Evidential DNN module. Additionally, each of these key components substantially contributes to enhancing the model's performance. The model's performance decreases depending on when we replace the backbones.

#### 4.6 Qualitative Results

Fig. 3 presents a visual comparison of qualitative results obtained from the OpenEAR method and various baseline approaches, using video samples from the EPIC-KITCHENS-55 dataset. This figure illustrates the enhanced capability of OpenEAR in accurately recognizing known actions and effectively rejecting

**Table 3.** Ablation study results on EPIC-KITCHENS-55.

Experiments		Action Acc		Object Acc		Union Acc	
		Top1	Top5	Top1	Top5	All Top1	All Top5
Ablate backbone	SwinT	0.4129	0.8989	0.3628	0.6506	0.1742	0.6015
	TimeS	0.4283	0.9057	0.3782	0.6612	0.1877	0.6112
Optimise the back		0.4601	0.9201	0.4129	0.6949	0.2300	0.6510
<b>Our Method</b>		<b>0.7074</b>	<b>0.9615</b>	<b>0.6603</b>	<b>0.8479</b>	<b>0.5080</b>	<b>0.8260</b>

									
<b>GT</b>	open (known)	container (known)	put (unknown)	container (known)	wash (known)	courgette (known)	cut (unknown)	courgette (known)	
<b>DEAR</b>	open container		open container		wash plate		wash plate		
<b>HTT</b>	open container		open container		wash courgette		wash courgette		
<b>OpenEAR</b>	Uncertainty	0.0002	0.0001	0.1100	0.0001	0.0003	0.0020	0.1200	0.0001
	Prediction	open	container	unknown	container	wash	courgette	unknown	courgette

**Fig. 3.** Qualitative results of the OpenEAR model and baseline models.

unknown ones, compared to the baseline models. While OpenEAR occasionally encounters recognition errors, these instances are accompanied by a notable level of uncertainty. This uncertainty quantification, a key feature of OpenEAR, adds a layer of reliability to the classification results, indicating the model’s confidence in its predictions. This aspect is particularly valuable in scenarios where discerning between known and unknown actions is crucial.

#### 4.7 Out-of-distribution Experiments

We tested OpenEAR’s performance under different OOD scenarios before and after uncertainty estimation was introduced, and reported the experimental results in Table 4. AU, OU, and AOU represent replacing parts of the test set with unseen samples from action unseen, object unseen, and unseen samples from action and object unseen, respectively. The Replacement Proportion represents how many samples in the test set are replaced by samples in the unseen subset. From the experimental results, it can be observed that, for Action Acc and Object Acc, our method shows varying degrees of improvement in Top1 after introducing uncertainty estimation in different OOD scenarios. The maximum improvement for Action Acc is 14.10%, and for Object Acc, it is 24.98%. As for Union Acc, our method exhibits different degrees of improvement in Top1 and Top5 after introducing uncertainty estimation. The average improvement for Top1 is 17.39%, and for Top5, it is 15.94%.

## 5 Conclusion

We introduced OpenEAR, a novel framework for open-set egocentric video action recognition, designed to overcome the limitations of traditional methods under closed-set assumption in open-world environments. Leveraging large-scale pre-trained models, OpenEAR excels in high-level semantic extraction and recognition predictions from egocentric videos by enhancing the network architecture. Its use of EDL for uncertainty estimation ensures the reliability of action recognition. By accurately identifying known actions and objects, and transparently demonstrating their credibility and confidence, OpenEAR uniquely balances accuracy and credibility, avoids the blindly overconfident recognition predictions, especially in identifying known actions and managing unknowns. This framework not only meets the dual requirements of accuracy and reliability but also provides more dependable action recognition results for practical applications. Future work will focus on expanding its application range and continually enhancing its overall performance and robustness.

**Acknowledgment.** This work was supported in part by the National Natural Science Foundation of China, No.:62376140, and No.:U23A20315; the Science and Technology Innovation Program for Distinguished Young Scholars of Shandong Province Higher Education Institutions, No.:2023KJ128, and in part by the Special Fund for distinguished professors of Shandong Jianzhu University.

**Table 4.** Performance comparison under different OOD settings.

Test Sets	Uncertainty	Replacement Proportion	Action Acc		Object Acc		Union Acc		
			Top1	Top5	Top1	Top5	All Top1	All Top5	
AU	×	25%	0.5365	0.7202	0.5962	0.8000	0.3837	0.6212	
	√		<b>0.6014</b>	<b>0.7973</b>	<b>0.8243</b>	<b>0.9257</b>	<b>0.5540</b>	<b>0.7568</b>	
OU	×		0.6798	0.9481	0.4933	0.6385	0.3837	0.6212	
	√		<b>0.5806</b>	<b>0.9306</b>	<b>0.7431</b>	<b>0.8194</b>	<b>0.5694</b>	<b>0.7777</b>	
AOU	×		0.5356	0.7202	0.4933	0.6385	0.3837	0.6212	
	√		<b>0.6054</b>	<b>0.8027</b>	<b>0.7279</b>	<b>0.8027</b>	<b>0.5578</b>	<b>0.7619</b>	
AU	×		50%	0.3510	0.4788	0.5125	0.7481	0.2471	0.4077
	√			<b>0.4265</b>	<b>0.5809</b>	<b>0.7426</b>	<b>0.9044</b>	<b>0.3824</b>	<b>0.5515</b>
OU	×	0.6327		0.9365	0.3192	0.4212	0.2471	0.4077	
	√	<b>0.6017</b>		<b>0.9237</b>	<b>0.5847</b>	<b>0.6780</b>	<b>0.4407</b>	<b>0.6356</b>	
AOU	×	0.3510		0.4788	0.3192	0.4212	0.2471	0.4077	
	√	<b>0.4915</b>		<b>0.6695</b>	<b>0.5847</b>	<b>0.6780</b>	<b>0.4407</b>	<b>0.6356</b>	

## References

1. Minlong Lu, Danping Liao, and Ze-Nian Li, “Learning spatiotemporal attention for egocentric action recognition,” in Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 4425–4434
2. Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boulton, “Toward open set recognition,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, pp. 1757–1772
3. Chuanxing Geng, Sheng-jun Huang, and Songcan Chen, “Recent advances in open set recognition: A survey,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, pp. 3614–3631
4. Wentao Bao, Qi Yu, and Yu Kong, “Evidential deep learning for open set action recognition,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13349–13358
5. Wentao Bao, Qi Yu, and Yu Kong, “Opental: Towards open set temporal action localization,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2979–2989
6. Murat Sensoy, Lance Kaplan, Federico Cerutti, and Maryam Saleki, “Uncertainty-aware deep classifiers using generative models,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 5620–5627
7. Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao, “Uniformer: Unified transformer for efficient spatiotemporal representation learning,” arXiv preprint [arXiv:2201.04676](https://arxiv.org/abs/2201.04676), 2022
8. Murat Sensoy, Lance Kaplan, and Melih Kandemir, “Evidential deep learning to quantify classification uncertainty,” in Advances in Neural Information Processing Systems, 2018, pp. 3183–3193
9. Glenn Shafer, “Dempster-shafer theory,” in Encyclopedia of Artificial Intelligence, 1992, pp. 330–331
10. Audun Jøsang, Subjective logic, Springer, 2016
11. Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al., “Scaling egocentric vision: The epic-kitchens dataset,” in Proceedings of the European Conference on Computer Vision, 2018, pp. 720–736
12. Yin Li, Miao Liu, and James M. Rehg, “In the eye of beholder: Joint learning of gaze and actions in first person video,” in Proceedings of the European Conference on Computer Vision, 2018, pp. 619–635
13. Yilin Wen, Hao Pan, Lei Yang, Jia Pan, Taku Komura, and Wenping Wang, “Hierarchical temporal transformer for 3d hand pose estimation and action recognition from egocentric rgb videos,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21243–21253
14. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022
15. Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu, “Video swin transformer,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3202–3211
16. D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price et al., “Rescaling egocentric vision,” arXiv preprint [arXiv:2006.13256](https://arxiv.org/abs/2006.13256), 2020

17. K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu et al., "Ego4d: Around the world in 3,000 hours of egocentric video," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 18995–19012
18. A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in Proceedings of the International Conference on Computer Vision, 2011, pp. 407–414
19. Nguyen, T.H.C., Nebel, J.C. and Florez-Revuelta, F., "Recognition of activities of daily living with egocentric vision: A review," Sensors, 2016, pp. 72
20. C. Dibiyadip and S. Fadime and M. Shugao and Y. Angela, "Opening the vocabulary of egocentric actions," Advances in Neural Information Processing Systems, 2024, pp. 33174–33187
21. Michael Land and Benjamin Tatler, "Looking and acting: vision and eye movements in natural behaviour," Oxford University Press, 2009
22. A. Bulling, J. A. Ward, H. Gellersen, and G. Troster, "Eye movement analysis for activity recognition using electrooculography," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, pp. 741–753
23. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," Advances in Neural Information Processing Systems, 2014, pp. 568–576
24. C. Li, S. Li, Y. Gao, X. Zhang, and W. Li, "A two-stream neural network for pose-based hand gesture recognition," IEEE Transactions on Cognitive and Developmental Systems, 2021, pp. 1594–1603
25. M. Liu, L. Ma, K. Somasundaram, Y. Li, K. Grauman, J. M. Rehg, and C. Li, "Egocentric activity recognition and localization on a 3d map," in Proceedings of the European Conference on Computer Vision, 2022, pp. 621–638
26. A. Furnari and G. M. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6252–6261
27. Y. Huang, M. Cai, Z. Li, F. Lu, and Y. Sato, "Mutual context network for jointly estimating egocentric gaze and action," IEEE Transactions on Image Processing, 2020, pp. 7795–7806
28. M. A. Arabacı, F.Ozkan, E. Surer, P. Jancovic, and A. Temizel, "Multi-modal egocentric activity recognition using audio-visual features," arXiv preprint [arXiv:1807.00612](https://arxiv.org/abs/1807.00612), 2018
29. S. Singh, C. Arora, and C. Jawahar, "First person action recognition using deep learned descriptors," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2620–2628
30. Yang, L.: Egocentric action recognition from noisy videos. The University of Tokyo, Diss. (2020)
31. F. Li and H. Wechsler, "Open set face recognition using transduction," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, pp. 1686–1697
32. A. Bendale and T. E. Boult, "Towards open set deep networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1563–1572
33. Z. Ge, S. Demnyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," arXiv preprint [arXiv:1707.07418](https://arxiv.org/abs/1707.07418), 2017
34. L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in Proceedings of the European Conference on Computer Vision, 2018, pp. 613–628
35. L. Ditria, B. J. Meyer, and T. Drummond, "Opengan: Open set generative adversarial networks," arXiv preprint [arXiv:2006.16241](https://arxiv.org/abs/2006.16241), 2020

36. P. Oza and V. M. Patel, "C2ae: Class conditioned auto-encoder for open-set recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2307–2316
37. R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4016–4025
38. X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional gaussian distribution learning for open set recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13480–13489
39. Tran, D., Snoek, J., Lakshminarayanan, B.: Practical uncertainty estimation and out-of-distribution robustness in deep learning. Technical Report, Google Brain (2020)
40. G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, 2021, pp. 1–38
41. D. Mandal, S. Narayan, S. K. Dwivedi, V. Gupta, S. Ahmed, F. S. Khan, and L. Shao, "Out-of-distribution detection for generalized zero-shot action recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9985–9993
42. A. Bendale and T. Boulton, "Towards open world recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1893–1902
43. S. N. Aakur and S. Kundu and N. Gunti, "Knowledge guided learning: Open world egocentric action recognition with zero supervision", *Pattern Recognition Letters*, 2022, pp. 38-45
44. S. Kuniaki and K. Donghyun and S. Kate, "Openmatch: Open-set semi-supervised learning with open-set consistency regularization", *Advances in Neural Information Processing Systems*, 2021, pp. 25956-25967
45. Y. Shu, Y. Shi, Y. Wang, Y. Zou, Q. Yuan, and Y. Tian, "Odn: Opening the deep network for open-set action recognition," in 2018 IEEE International Conference on Multimedia and Expo. IEEE, 2018, pp. 1–6
46. R. Krishnan, M. Subedar, and O. Tickoo, "Bar: Bayesian activity recognition using variational inference," arXiv preprint [arXiv:1811.03305](https://arxiv.org/abs/1811.03305), 2018
47. M. Subedar, R. Krishnan, P. L. Meyer, O. Tickoo, and J. Huang, "Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6301–6310
48. R. Krishnan, M. Subedar, and O. Tickoo, "Specifying weight priors in bayesian deep neural networks with empirical bayes," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 4477–4484
49. P. P. Busto, A. Iqbal, and J. Gall, "Open set domain adaptation for image and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, pp. 413–429
50. A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058
51. B. Charpentier, D. Zügner, and S. Gunnemann, "Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts," *Advances in Neural Information Processing Systems*, 2020, pp. 1356–1367
52. W. Shi, X. Zhao, F. Chen, and Q. Yu, "Multifaceted uncertainty estimation for label-efficient deep learning," *Advances in Neural Information Processing Systems*, 2020, pp. 17247–17257



53. F. Kraus and K. Dietmayer, “Uncertainty estimation in one-stage object detection,” IEEE Intelligent Transportation Systems Conference, 2019, pp. 53–60
54. W. Bao, Q. Yu, and Y. Kong, “Uncertainty-based traffic accident anticipation with spatio-temporal relational learning,” in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 2682–2690
55. A. Amini, W. Schwarting, A. Soleimany, and D. Rus, “Deep evidential regression,” Advances in Neural Information Processing Systems, 2020, pp. 14927–14937



# Temporal Divide-and-Conquer Anomaly Actions Localization in Semi-supervised Videos with Hierarchical Transformer

Nada Osman<sup>1</sup>(✉) and Marwan Torki<sup>2</sup>

<sup>1</sup> University of Padova, Padua, Italy  
nadasalahmahmoud.osman@phd.unipd.it

<sup>2</sup> Alexandria University, Alexandria, Egypt  
mtorki@alexu.edu.eg

**Abstract.** Anomaly action detection and localization play an essential role in security and advanced surveillance systems. However, due to the tremendous amount of surveillance videos, most of the available data for the task is unlabeled or semi-labeled with the video class known, but the location of the anomaly event is unknown. In this work, we target anomaly localization in semi-supervised videos. While the mainstream direction in addressing this task is focused on segment-level multi-instance learning and the generation of pseudo labels, we aim to explore a promising yet unfulfilled direction to solve the problem by learning the temporal relations within videos in order to locate anomaly events. To this end, we propose a hierarchical transformer model designed to evaluate the significance of observed actions in anomalous videos with a divide-and-conquer strategy along the temporal axis. Our approach segments a parent video hierarchically into multiple temporal children instances and measures the influence of the children nodes in classifying the abnormality of the parent video. Evaluating our model on two well-known anomaly detection datasets: UCF-crime and ShanghaiTech proves its ability to interpret the observed actions within videos and localize the anomalous ones. Our proposed approach outperforms previous works relying on segment-level multiple-instance learning approaches while reaching a promising performance compared to the more recent pseudo-labeling-based approaches. Our code is available at our [GitHub Repo](#).

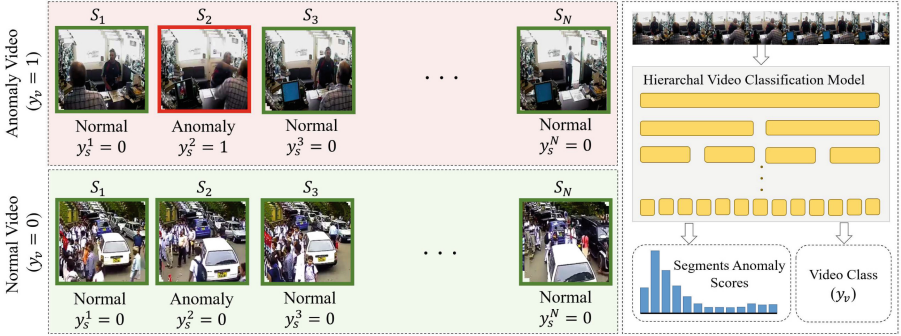
**Keywords:** Anomaly Actions Localization · Anomaly Detection · Weakly Supervised Learning · Semi-Supervised Learning · Hierarchical Modeling · Transformers

## 1 Introduction

Surveillance represents the backbone of almost all security systems; however, extracting important events, particularly anomalies, from the vast pool of collected videos is a time-consuming and exhaustive task. There arises a critical need for an intelligent system capable of accurately and autonomously extracting and localizing events of interest. The main challenge in training such a localization model lies in the lack of

supervision, as the massive amount of collected data for this task is unsupervised or weakly supervised. Consequently, the employed model must be able to decode event sequences, extract embedded relations, and identify potential outlier behaviors. Many prior works have tackled this task, some in a fully unsupervised approach [4, 11], but mostly in a semi-supervised approach [8, 10, 20, 28, 30]. In the semi-supervised setting, each video is labeled as normal or anomalous, yet the specific location of the anomaly segment within the video is unspecified. The standard protocol, in this case, operates at the segment level, aiming to maximize the hidden representation gap between normal and anomalous segments [20, 28, 30]. To further improve the learning process in weakly supervised settings, the research community has started to give more attention to the generation and refinement of per-segment pseudo labels [1, 8, 10, 13, 27, 29]. Alternatively, recent attention has been directed towards image reconstruction approaches, wherein models are trained to reconstruct normal videos or frames [3, 25, 26]. Subsequently, these trained models are repurposed to distinguish between high-quality reconstructions, indicative of normal frames, and poor-quality reconstructions, indicative of anomaly frames. These anomaly detection techniques are segment-level or frame-level approaches, often overlooking per-video classification. However, it is reasonable that decoding the relative dependencies within videos should allow for event understanding and anomaly localization. This approach would offer several advantages: 1) Reformulating the anomaly detection task from segment-based to video-based is more compatible with real-world surveillance applications, enabling the processing of longer videos rather than just short segments. 2) It enhances explainability and human understanding of the anomaly event and its temporal context. 3) Compared to pseudo-labeling, it provides a faster end-to-end training process. 4) Hypothetically, it should offer higher generalization ability than pseudo-labeling techniques, which are tailored to specific datasets and may inherit dataset specifics and noise. Therefore, in this work, we explore this direction, proposing a novel approach to utilize per-video classification for anomaly localization.

If a model can distinguish videos containing anomalies from normal videos, the model’s hidden representation should inherently contain sufficient information about the anomaly locations. We propose a temporal divide-and-conquer transformer-based model to classify the normality of a parent video and its children segments in a hierarchical approach. We extract potential anomaly segments based on the aggregated classifications of the model throughout the hierarchical levels, integrated with the corresponding activation maps. As shown in Figure 1, a video is split into  $N$  temporal segments, where a normal video does not contain any anomaly events, while an abnormal video must contain at least one anomalous action. Unlike previous works, our objective is to solve two tasks: the per-video classification task, predicting  $y_v$ , and the per-segment classification, predicting  $y_s^i$  for each segment  $i$  in the video. The per-segment classification is fully unsupervised, based on the information extracted during the per-video classification. As shown in Figure 1, our model processes the input video in a hierarchical manner, where each level of the hierarchy aims to measure the abnormality of the included video patches, reaching up to the final level representing the fine-grained segments. Therefore, the proposed model produces two outputs: a high-level



**Fig. 1.** Each video is split into  $N$  segments. A normal video ( $y_v = 0$ ) contains only normal segments ( $y_s^i = 0, \forall i \in [1 : N]$ ). While an anomaly video ( $y_v = 1$ ) contains at least one anomaly segment ( $y_s^i = 1, \exists i \in [1 : N]$ ). Our approach employs a hierarchical transformer model to classify the abnormality of the whole video, in addition to producing abnormality scores for the individual segments. This approach differs from previous works that overlook the context of the entire video and classify individual segments independently. Prior methods typically apply multiple-instance learning (MIL) to distinguish normal segments from anomalies, irrespective of their context, or generate per-segment pseudo labels to compensate for the lack of supervision.

per-video classification and per-segment abnormality scores. Our evaluation results provide promising insights into the effectiveness of our proposed approach in understanding the observed video, extracting the temporal relations, and identifying the anomaly events. The main contributions of our work can be summarized as follows:

1. We revise the segment-wise anomaly detection task, transforming it into anomaly localization within videos, allowing the learning procedure to benefit from per-video classification in detecting the anomaly segments in a semi-supervised manner.
2. We propose our temporal divide-and-conquer transformer-based model that aims to weigh the abnormality of various temporal patches of the video hierarchically in order to provide a fine-grained aggregated estimation of the abnormality of the detected events.
3. We evaluate our model on two well-known datasets for anomaly detection and conduct different ablation experiments.

## 2 Related Work

### 2.1 Anomaly Detection

Anomaly detection is an important task that has been always gaining attention in computer vision. The task takes three primary learning formats: Fully-supervised learning, where ground truth labels are available for both normal and anomaly actions [15, 26]; Semi-supervised learning, having ground truth labels for entire videos but lacks annotations for anomaly segments within those videos [3, 8, 10, 20, 28–30]; and Fully-unsupervised learning, which operates without any ground truth labeling [4, 11]. While

full supervision offers a direct path to problem-solving, the tremendous process of annotating large datasets makes it impractical. Conversely, unsupervised learning bypasses the need for data annotations but introduces heightened complexity to the learning process. Consequently, significant attention has shifted towards the semi-supervised learning format, where models can leverage video annotations to enhance the recognition of anomaly segments within those videos.

In the semi-supervised setting, previous works primarily applied multiple instance learning (MIL), where the model pairs one normal video with another anomaly video and trains the model to maximize the representation gap between the two segments. The leading work in this category is [20], which aimed to maximize the distinction between the highest-scoring segments within the positive and negative bags, thereby maximizing the inter-bag distance. Building upon this concept, [3] introduced a novel approach by concurrently minimizing intra-bag distance while maximizing inter-bag distance. Additionally, in [26], MIL is once again employed, this time augmented with an attention mechanism aimed at ranking segments within each bag, with the objective of accentuating the disparity between the highest-ranking segments across both bags.

Another leading technique for the semi-supervised learning approach is pseudo-labeling. This approach involves generating pseudo-labels for unlabeled segments and subsequently training the model in a fully-supervised manner using these generated labels. In [29], a method is proposed where an action classifier module is trained for each segment. This module takes segment images and the video's label as inputs, iteratively refining and purifying the predicted segment classifications. Conversely, in [8, 10, 22] [10], multiple instance learning (MIL) is employed to generate pseudo-labels, which are utilized to train the classifier model. With the rising attention to pseudo-labeling approaches, some recent works are dedicated to the generation, refinement, and cleaning process of pseudo-labels, as in [1, 27]. In contrast to these works, we reformulate the problem into anomaly localization within videos, proposing a model that can interpret the temporal axis along input videos, extracting anomaly events.

## 2.2 Class Activation Maps Learning

Class activation maps (CAM) learning is a popular technique in computer vision. It is mainly used in image-based object detection, where CAM-based techniques are widely used in scoring objects within images for object detection and localization [6, 12, 18, 24]. Recently, the application of CAM techniques to videos started to emerge. For example, in [2], temporal max pooling is proposed to aggregate per-frame CAMs for video object localization. Another interesting application of CAM learning in videos is [16], where optical flow-based CAM is utilized for weakly-supervised segmentation. In the domain of anomaly detection, [21] proposed the utilization of CAM to generate a form of pseudo labeling, integrated into a second learning phase with MIL to create two groups of training segments: positive and negative. The final classification in this approach employs K-nearest neighbor to the positive and negative groups. Similarly, in [22], segment activations, represented by features magnitude, are utilized to generate top-k normal/anomaly segments that are then used to train a per-segment classifier for

anomaly detection. Our objective in this work is to generate different types of activation maps, which are then fused together to yield an attentive estimation of the abnormality associated with the events observed within the classified video.

### 2.3 Temporal Hierarchical Modeling

Understanding and extracting temporal dynamics from videos and real-world time series pose significant challenges. Consequently, techniques like temporal multi-scale and hierarchical modeling are important in processing temporal data. For instance, in [9], the proposed model operates across two temporal scales, slow and fast, resulting in more robust representation extraction. Numerous studies have adopted hierarchical modeling methods for real-world temporal data, such as [19], or for video tasks, such as action recognition [7, 14]. Therefore, in our work, we utilize hierarchical modeling along the temporal axis to leverage temporal information extracted from input videos at varying temporal resolutions using a transformer-based model.

## 3 Temporal Divide-and-Conquer Approach

We consider two classification tasks: per-video classification and per-segment classification, employing our temporal divide-and-conquer model depicted in Figure 2. The model takes a video as input, which is divided into a sequence of  $N$  segments. Each segment represents an action or event and consists of a set of frames. Raw images within each segment are projected into the feature space using our double-scale features extractor (DS- $\Phi$ ) module, followed by processing through our hierarchical transformer layers.

### 3.1 Double Scale Features Extractor

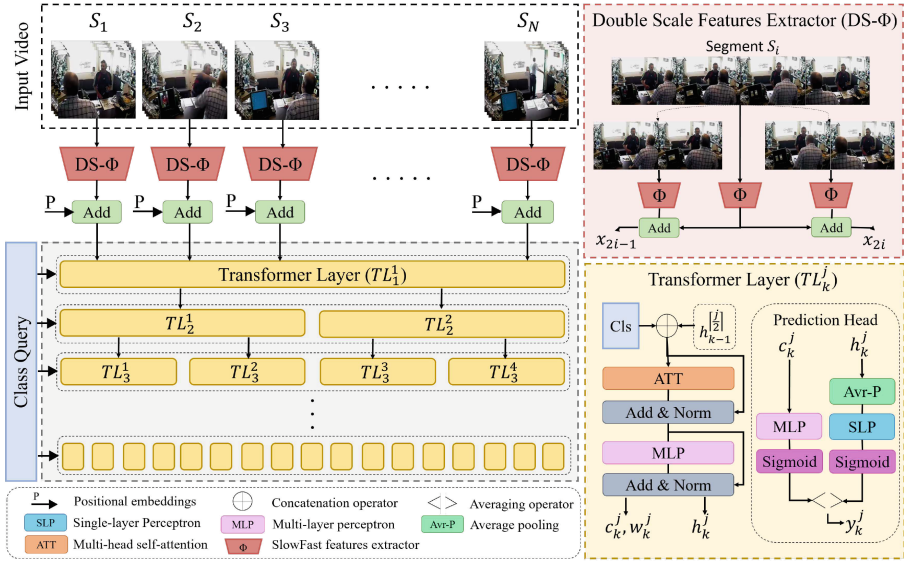
Our DS- $\Phi$  module is designed to enhance the temporal resolution of input segments, effectively doubling the amount of information extracted during clip processing. Given that anomaly events within a segment may exhibit varying degrees of significance across different parts of the segment, our module enables both collective and divided processing of segments. This approach allows the extraction of more focused features, leading to a deeper understanding of the underlying events. Each segment  $S_i$  is split into  $S_i^1$  and  $S_i^2$ , with the raw images of  $S_i$ ,  $S_i^1$ , and  $S_i^2$  projected into the feature space using  $\Phi$ , as in (1), applying a feature mapping with multi-layer perceptron over the extracted features ( $Feat$ ) by a pre-trained video-framework.

$$\Phi(S) = MLP(Feat(S)) \quad (1)$$

Subsequently, DS- $\Phi$  allows the generation of the double temporal resolution by producing two feature vectors  $\mathbf{x}_{2i-1}$  and  $\mathbf{x}_{2i}$  from each segment  $S_i$ , as in (2), where  $\mathbf{x} \in \mathbb{R}^{1 \times D}$  and  $D$  is the features size. This approach allows for extending the temporal length to its double ( $2N$ ).

$$\mathbf{x}_{2i-1} = \Phi(S_i) + \Phi(S_i^1) \quad \mathbf{x}_{2i} = \Phi(S_i) + \Phi(S_i^2) \quad (2)$$

Our model is not fixed to a specific pre-trained feature extractor; however, we consider two pre-trained models as our backbones: I3D [5], and Slowfast [9].



**Fig. 2.** Our divide-and-conquer transformer-based model operates by taking the segmented video as input, where the video is divided into  $N$  segments. These segments undergo feature extraction using our **Double Scale Features Extractor (DS- $\Phi$ )** module. Subsequently, the extracted features are passed to the **hierarchical transformer layers** for classification. At the first level ( $TL_1^1$ ), the model generates video classification ( $y_1^1 = y_v$ ), and at each subsequent level ( $T_k$ ), it produces sup-video classification  $y_k^j, \forall j \in [1, 2, 3, \dots, 2^{k-1}]$ .

### 3.2 Hierarchical Transformer Layers

Our divide-and-conquer approach begins with the coarse-grained task of per-video classification, iteratively breaking down the task into smaller sub-tasks (clips) until reaching the fine-grained task of per-segment classification. The features of the complete sequence  $\mathbf{x} \in \mathbb{R}^{2N \times D}$  are combined with positional embeddings  $P \in \mathbb{R}^{2N \times D}$  to retain temporal causality, forming  $(\mathbf{x} + P)$ . Subsequently, this combined input is forwarded to the first level of the hierarchical transformer layers and processed through the hierarchy as shown in Figure 2.

At each prediction level  $k \in [1, K]$ , where  $K$  denotes the length of the hierarchy, the model binary-divides the received signals  $h_{k-1}$  from the preceding level  $k - 1$  into  $2^{k-1}$  splits, where we set the initial input signal  $h_0$  to  $\mathbf{x} + P$ , as mentioned above. To better describe the processing flow throughout our hierarchical model, we define the following for a single split  $P_k^j$  within the  $k^{th}$  level in the hierarchy, where  $j \in [1, 2^{k-1}]$ . Specifically, a split  $P_k^j$  represents a cut patch from the input video that has been processed up to level  $k$ . This patch is then handled by a self-attention transformer layer  $TL_k^j$ , which processes  $P_k^j$  and extracts a new hidden representation. The input to  $TL_k^j$  is composed of two concatenated parts, as described in (3), combining a fixed class query input ( $Cls$ ), along with the hidden representation received from the preceding level  $k - 1$  and the

parent split  $P_{k-1}^{\lceil \frac{j}{2} \rceil}$ , defined as  $h_{k-1}^{\lceil \frac{j}{2} \rceil}$ . The  $\oplus$  in (3) denotes the sequence concatenation operation.

To this end, each transformer layer  $TL_k^j$  produces three outputs for  $P_k^j$ , as given in (9):  $c_k^j \in \mathbb{R}^{1 \times D}$ , representing the class *Cls* encoding at the layer;  $w_k^j \in \mathbb{R}^{1 \times 2N}$ , denoting the attention weights inside the class query; and  $h_k^j \in \mathbb{R}^{\frac{N}{2^{k-1}} \times D}$ , representing the encoding vectors of the segments present in video patch  $P_k^j$ . The hidden encoding of the segments  $h_k^j$  is next passed to the proceeding level  $k+1$  for finer-grained processing of the segments, reaching up to the bottom  $K^{th}$  level.

$$c_k^j, w_k^j, h_k^j = TL(Cls \oplus h_{k-1}^{\lceil \frac{j}{2} \rceil}), \quad \forall k \in [1, K], \quad \forall j \in [1 : 2^{k-1}] \quad (3)$$

### 3.3 Prediction Head

Finally, following each transformer layer, a prediction head generates an abnormality classification. The transformer layer at the first level  $TL_1^1$  produces the per-video classification, while each subsequent layer  $TL_k^j$  generates an estimation of the abnormality for the video split processed at that layer, denoted as  $y_k^j$  as depicted in Figure 2. To enable the model assessing the influence of segments in predicting each sup-video (split), two prediction approaches are employed: Utilizing the generated class encoding at the layer ( $c_k^j$ ) to produce the classification  $(y_k^j)_c$ , as shown in (4); and employing a sigmoid classifier head, with a single layer perceptron (SLP), on the average-pooled segments encoding vectors  $h_k^j$ , as illustrated in (5), generating  $(y_k^j)_h$  and allowing for acquiring an activation map of the enclosed segments.

$$(y_k^j)_c = Sigmoid(MLP(c_k^j)) \quad (4)$$

$$(y_k^j)_h = Sigmoid(SLP(AveragePooling(h_k^j))) \quad (5)$$

The final prediction is computed as the average of  $(y_k^j)_c$  and  $(y_k^j)_h$ , given by:

$$y_k^j = Average((y_k^j)_c + (y_k^j)_h) \quad (6)$$

### 3.4 Localization Approach

The model breaks the video prediction into a set of sub-predictions, where we aim to measure the influence of a segment  $S_i$  in all of its corresponding sub-predictions. To capture such influence, we rely on three measuring factors, as illustrated in Figure 3:

1. The abnormality prediction in the corresponding patches across the different levels ( $p_i$ ); the probability of  $S_i$  to be an anomaly segment is monotonically increasing with the corresponding probability of a parent clip. Therefore, this probability is measured as the averaged predictions of the parent clips across the  $K$  levels, given by:



$$p_i = \underset{k \in [1, K]}{\text{Average}} \ y_k^j, \quad j = \lceil \frac{i}{2^{k-1}} \rceil \tag{7}$$

2. The activation effect of  $S_i$  in the prediction of parent patches ( $a_i$ ); The higher the activation of the segment in an anomaly class, the higher its probability of being an anomaly. The averaged activation across the levels is given by:

$$a_i = \underset{k \in [1, K]}{\text{Average}} \ h_k[i] \tag{8}$$

Where  $h_k$  is the stacked, average-pooled hidden representations of the  $2^{k-1}$  transformer layers at level  $k$ , and  $h_k[i]$  is the corresponding representation of  $S_i$ .

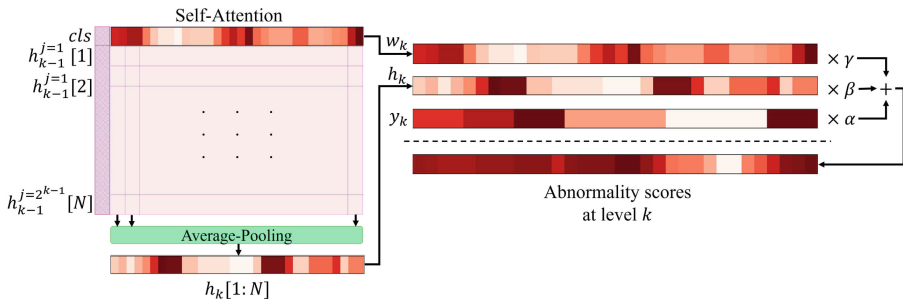
3. The corresponding attention weights of the segment  $S_i$  in the class query ( $t_i$ ); Again, the higher attention given to a segment during the anomaly prediction ( $y_k^j$ )<sub>c</sub>, the greater its influence and the higher its probability of being an anomaly. Therefore the attention maps of the class query in the corresponding splits are averaged across the levels to yield the attentive weight of  $S_i$  in the prediction:

$$t_i = \underset{k \in [1, K]}{\text{Average}} \ w_k[i] \tag{9}$$

Where,  $w_k$  is the concatenated attention weights across the employed attentions at level  $k$ , and  $w_k[i]$  is the weight of the corresponding segment  $S_i$ .

Based on these three factors,  $p_i$ ,  $a_i$ , and  $t_i$ , an aggregation of the abnormality estimation score  $e_i$  of  $S_i$  is calculated as the weighted average of the normalized factors, as specified in (10). Here,  $\alpha$ ,  $\beta$ , and  $\gamma$  represent the weighting parameters. Finally, considering the tendency of anomaly segments to occur in clusters within the video, we smooth the estimated probabilities across the video using a moving average. Subsequently, we apply spike filtering to eliminate potential outlier scores.

$$e_i = \alpha p_i + \beta a_i + \gamma t_i \tag{10}$$



**Fig. 3. Visualization of our localization Approach.** Assuming the localization is conducted at level  $k$ ,  $w_k$  is obtained from the attention weights of the class query inside our self-attention layers,  $h_k$  is the averaged-pooled encodings produced by the transformer, and  $y_k$  is the stacked sub-predictions at  $k$ . The estimated abnormality is computed as in Equation (10).

## 4 Experimental Results

### 4.1 Dataset

We evaluate our model on two datasets: UCF-Crime dataset [20] and ShanghaiTech [17]. UCF-Crime dataset comprises 1,900 surveillance videos spanning over 128 hours, with 13 distinct anomaly behaviors such as abuse, assault, accident, fighting, robbery, etc. On the other hand, ShanghaiTech is a smaller dataset, comprising 437 videos, with 130 anomaly videos and 307 normal videos. The videos are partitioned into 32 segments each, with ground truth per-segment class labels provided exclusively for the segments in the test split of the data.

### 4.2 Evaluation Metrics

Following the established evaluation protocol for this task [30], we utilize the Area under the ROC Curve (AUC-ROC) metric to assess the localization performance of anomaly segments. Additionally, we report accuracy and F1-score metrics to measure the model’s ability to distinguish between normal and abnormal segments.

### 4.3 Implementation details

For both of our evaluation datasets, we set the number of segments  $N$  to 32 segments. The extracted feature sizes of the pre-trained video frameworks are 2048 and 2304 for I3D and SlowFast, respectively, while the mapped features size  $D$  is set to 256 and 288. Each transformer layer includes 4 self-attention heads, and the hierarchical model consists of  $K = 6$  levels. The Multi-Layer Perceptron (MLP) module inside the prediction head consists of three layers with sizes 128, 64, and 32. The dropout rate for all model layers is set to 0.1. The model is trained for 100 epochs using a Stochastic Gradient Descent (SGD) optimizer, with a learning rate of 0.01, and binary cross-entropy loss function. During training, all children split within the hierarchical model inherit their anomaly class label from the parent video. During inference, our localization approach is applied using the weighting parameters  $\alpha$ ,  $\beta$ , and  $\gamma$ , configured to 0.9, 0.05, and 0.05, for UCF-crime. While 0.65, 0.3, and 0.05 are used for ShanghaiTech.

**Table 1. Comparison with the state-of-the-art works on UCF-Crime.** For a fair comparison, we separate the performance based on the application of pseudo-labeling (PL). Our model uniquely overlooks both MIL and pseudo-labeling techniques. Instead, it employs a hierarchical measuring technique for abnormality weights of the per-video segments, outperforming all previous works w/o PL. However, the best performances are in favor of the utilization of tailored pseudo labels for the dataset, with a comparable performance of our HCAM-former with most PL-works.

	MIL	Segments Activation	Features Extractor	Classifier	AUC	
					w/o PL	PL
MIL (2018) [20]	✓	✗	C3D	FC	75.41	-
TCN-IBL (2019) [28]	✓	✗	C3D	TCN	78.66	-
MIL-MA (2019) [30]	✓	✗	PWCNet	TAN	79.00	-
GC-LNC (2019) [29]	✗	✗	TSN	GCN	70.87	82.12
CPL (2020) [21]	✓	✓	BN-Inception	ResNet+KNN	-	79.31
MIL-MIST (2021) [10]	✓	✗	I3D	Attention	73.33	82.30
MIL-AUG (2022) [8]	✓	✗	SlowFast	FC	79.37	81.24
RTFM (2022) [22]	✓	✓	I3D	PDC	-	84.30
MSL (2022) [13]	✓	✗	VideoSwin	Transformer	-	85.62
CU-Net (2023) [27]	✓	✗	I3D	FC	-	<b>86.22</b>
C2FPL (2024) [1]	✓	✗	I3D	FC	72.70	85.50
HCAM-former (Ours)	✗	✓	I3D SlowFast	H-Transformer	<b>78.30</b> <b>79.47</b>	- -

#### 4.4 Results

We conduct a comparative analysis of our model against state-of-the-art works in anomaly detection. Table 1 presents a comparison on the UCF-Crime dataset, highlighting the structural differences between models to ensure a fair evaluation. This includes whether the learning process employs multiple-instance learning, benefits from segment activation, the feature extractor used, and the final classification module of each detection methodology. Additionally, we differentiate the performance of the models based on the application of pseudo-labeling. As shown in the table, our HCAM-former outperforms all other techniques when pseudo-labeling is excluded, underscoring the efficacy of our method in interpreting observed events and distinguishing anomalies. However, while our model achieves promising results, it generally falls behind the performance of pseudo-labeling approaches. These methods handle weakly supervised data by generating estimated labels for refinement and use as ground truth, and such iterative learning techniques enhance anomaly pattern recognition. As mentioned earlier, our aim is to provide a more generic solution that is not as dataset-specific as the pseudo-labeling technique. It is noteworthy that our model still outperforms the CPL [21] approach, which utilizes segments CAM (only hidden representation CAM  $a_i$  in our approach), in addition to employing both MIL and pseudo-labeling techniques. Similarly, Table

**Table 2. Comparison with the state-of-the-art works on ShanghaiTech.** Again, the HCAM-former outperforms all w/o PL models while achieving comparable performance to the best PL works.

	MIL	Segments Activation	Features Extractor	Classifier	AUC	
					w/o PL	PL
MIL (2018) [20,22]	✓	✗	C3D	FC	85.33	-
GC-LNC (2019) [29]	✗	✗	TSN	GCN	80.83	84.44
AR-Net (2019) [23]	✓	✗	I3D	FC	91.24	-
MIL-MIST (2021) [10]	✓	✗	I3D	Attention	-	94.38
RTFM (2022) [22]	✓	✓	I3D	PDC	-	97.21
MSL (2022) [13]	✓	✗	VideoSwin	Transformer	-	<b>97.32</b>
HCAM-former (Ours)	✗	✓	I3D	H-Transformer	<b>93.29</b>	-

2 reports the results on the ShanghaiTech dataset. Again, our model achieves state-of-the-art performance when excluding pseudo-labels while reaching a compatible performance with the PL approaches and even outperforms the GC-LNC(PL) technique.

The provided comparison results prove the effectiveness of our proposed approach in identifying anomaly events from normal ones solely based on the temporal progression of events within the parent context. Although it has a degraded performance compared to the best pseudo-labeling models, our model provides a more generic end-to-end solution that overlooks the tailoring of pseudo-labels for the evaluation datasets, which is a promising direction with respect to real-world applications.

#### 4.5 Ablation Study

We conducted an ablation study to assess the impact of different components of our proposed model on per-video classification performance on UCF-Crime, as summarized in Table 3. Since the model’s ability to recognize videos with anomaly events from normal videos depends on its capability to extract anomaly behaviors within the videos, achieving high per-video performance is crucial to ensure the accuracy of the estimated segment scores by the model. This ablation study examines three main components of our model: 1) The hierarchical structure (comparing  $K = 1$  against  $K = 6$ ), where  $K = 1$  denotes that only the first level of HCAM-former is trained and used for inference depending on  $a_i$  and  $t_i$ . While  $K = 6$  denotes the utilization of the whole hierarchy. 2) The inclusion of the class query in the transformer layers. 3) The double-scale features extractor module DS- $\Phi$ .

The hierarchical structure has the most significant impact, contributing to an approximately 3% increase in the F1-score and 9% in AUC, compared to the model without the hierarchy. Additionally, the class query and the DS- $\Phi$  module notably improve the model’s performance. As a result, the highest-performing configuration is achieved by

**Table 3. Ablation on per-video classification performance (UCF-Crime).** When  $K = 1$ , the model operates without the hierarchical structure. In this configuration, only the first transformer layer of our model is utilized to make predictions and measure the abnormality scores of the segments.

Hierarchical Transformer		Class	DS- $\phi$	ACC	AUC	F1
$K = 1$	$K = 6$	Query				
✓	✗	✗	✗	83.97	83.89	83.09
✓	✗	✓	✗	85.02	84.83	83.40
✓	✗	✓	✓	85.02	84.90	83.89
✗	✓	✓	✓	<b>87.12</b>	<b>92.44</b>	<b>86.74</b>

**Table 4. Ablation on the localization performance (UCF-Crime).** Where,  $a_i$  denotes the class activation maps in (8),  $t_i$  is the attention weights of the class query in (9), and  $p_i$  is probability estimation in (7).

Hierarchical Transformer		$a_i$	$t_i$	$p_i$	ACC	AUC	F1
$K = 1$	$K = 6$						
✓	✗	✓	✗	✗	<b>73.73</b>	72.83	40.11
✓	✗	✗	✓	✗	70.09	72.28	38.17
✓	✗	✓	✓	✗	73.18	73.87	<b>40.65</b>
✗	✓	✓	✓	✓	66.54	<b>79.47</b>	37.78

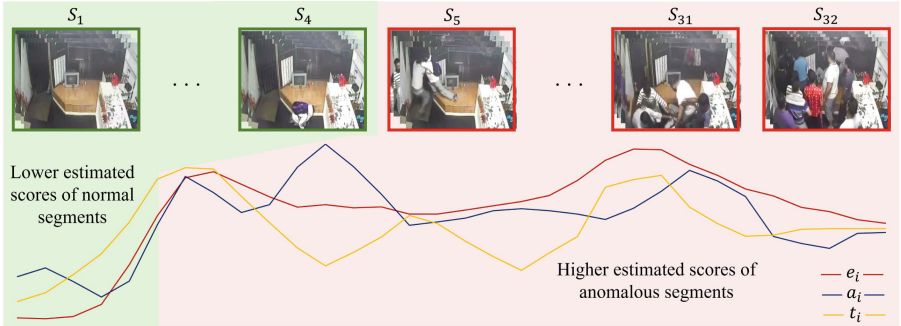
including all components of our model.

In Table 4, we ablate our localization technique. By excluding hierarchical predictions, we compare the localization performance using the class activation of the first transformer layer in the hierarchy ( $a_i$ ) and the attention weights of the class query of the same layer, both individually and integrated. The results in the table indicate that CAM estimation slightly outperforms attention weights in terms of localization. However, combining both estimations yields even better localization performance. Integrating hierarchical predictions as abnormality probability estimation achieves the highest localization performance, with an improvement of approximately 5% in AUC. As can be noticed, when the AUC metric is increased, accuracy and F1 score suffer a bit of degradation. This could be explained by the severe imbalance in the number of anomaly segments compared to normal ones. Therefore, maximizing the number of correctly localized anomalies leads to an increase in the number of misclassified normals. However, correctly localizing anomalies presents a higher priority, and therefore, the AUC metric is more relevant for the task.

Finally, Table 5 evaluates the impact of the feature extractor on anomaly localization performance. The results favor the SlowFast feature extractor over the I3D features, even when the number of heads in our transformer layers is increased to 8 to achieve

**Table 5. Ablation on the features extractor (UCF-Crime).** The ACC, AUC, and F1 are reported for both per-video classification and per-segment classification.

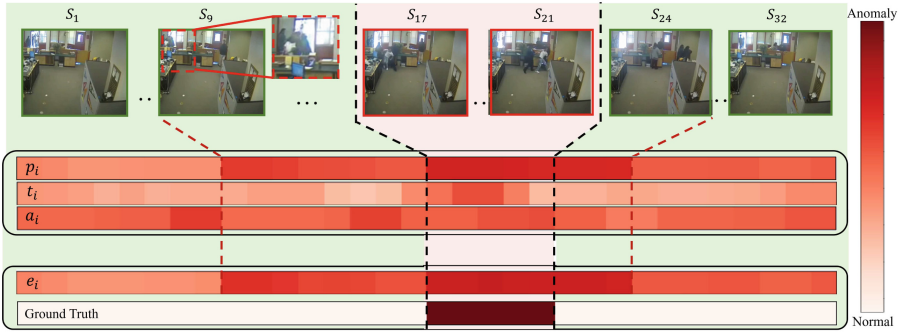
Features	Per-Video			Per-Segment		
	ACC	AUC	F1	ACC	AUC	F1
I3D	80.21	77.6	82.9	65.27	78.30	23.70
SlowFast	87.12	92.44	86.74	66.54	79.47	37.78

**Fig. 4. Qualitative example on accurate anomaly localization from UCF-Crime.** Anomaly score estimations are provided for the anomaly action "Assault", which begins at segment  $S_5$  and continues until the final segment  $S_{32}$ . Here,  $e_i$  represents the score estimated in (10),  $a_i$  denotes the segment's activation as defined in (8), and  $t_i$  refers to the attention weights described in (9).

acceptable performance with I3D features. This finding demonstrates the superior capability of the SlowFast framework to interpret spatial and temporal information in the input images, enabling our model to better understand and localize anomaly events in the videos.

#### 4.6 Qualitative Examples

Figure 4 illustrates the estimated anomaly scores for segments within an anomaly video featuring an "Assault" event. The video begins with normal events during the first segments [ $S_1 - S_4$ ]; then the anomalous event starts at  $S_5$ . The plotted anomaly scores for all three abnormality measuring factors exhibit lower values at the beginning, gradually increasing as the anomaly action unfolds. The aggregated anomaly estimation  $e_i$  yields smoother estimations across the segments, benefiting from the fusion of the used factors. While both the activation map  $a_i$  and attention weights  $t_i$  show slightly decreased scores during the anomaly event, they remain higher than those of the normal segments. Towards the end of the video, as the "Assault" event subsides, the anomaly estimation begins to decrease. This demonstrates the model's capability to interpret events depicted in the segments and distinguish anomalies, relying solely on the information gained during video classification.



**Fig. 5. Qualitative example on slightly misled anomaly localization from UCF-Crime.** The heat maps illustrate anomaly score estimations for the "Arrest" event, which initiates at segment  $S_{17}$  and extends through segment  $S_{21}$ . The illustrated heat maps are  $p_i, t_i, a_i$ , the aggregated estimation  $e_i$ , and the ground truth labeling of the segments.

The qualitative example depicted in Figure 5 aims to illustrate instances where the model may misinterpret anomaly segments. To provide a more sensitive analysis, heat maps are employed for visualization, where an "Arrest" event is occurring from segment  $S_{17}$  to  $S_{21}$ . Initially, all abnormality measures exhibit relatively lower scores during the video's early segments, indicating normalcy. However, starting from segment  $S_9$ , scores begin to rise, particularly for  $p_i$  and  $a_i$ . Notably, an event at  $S_9$  depicts a violent interaction in the left corner of the frame, leading to elevated abnormality scores for segments  $[S_9 - S_{16}]$ . Then, as the actual "Arrest" event unfolds, the model accurately assigns the highest abnormality scores; however, the highest scores are persisted until segment  $S_{24}$ . This is due to another misleading event featuring multiple individuals gathered around the arrest location, contributing to the sustained high scores beyond the actual event. It is worth noting that throughout the observation, attention scores provided by  $t_i$  closely align with the ground truth, yet they tend to emphasize only the most significant segments, potentially overlooking some anomalous events.

From both examples, it is evident that the factor with the most effective capability in recognizing anomaly events is the estimated probability using our hierarchical predictions.

## 5 Conclusion

In this work, we tackle the challenge of anomaly detection in weakly-supervised datasets. We redefine the anomaly detection task from per-segment to per-video classification, leveraging the temporal progression of videos to localize anomaly events effectively. Our proposed temporal divide-and-conquer hierarchical transformer model surpasses state-of-the-art non-pseudo-labeling methods and achieves promising results compared to tailored pseudo-labeling approaches. These findings demonstrate the promising capability of our model to process temporal video contexts, comprehend

observed events, and correctly localize anomalies. Although the current performance of our proposed approach does not yet match the higher performance achieved by tailoring pseudo-labels for the evaluation datasets, we explore a more explainable and general solution of video-level anomaly localization, showcasing the promising capability of such a technique in interpreting and localizing abnormal events. Therefore, our future work will focus on enhancing video-level anomaly localization, allowing for a more in-depth study of its performance, explainability, and generality compared to pseudo-labeling approaches.

## References

1. Al-Lahham, A., Tastan, N., Zaheer, M.Z., Nandakumar, K.: A coarse-to-fine pseudo-labeling (c2fpl) framework for unsupervised video anomaly detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6793–6802 (2024)
2. Belharbi, S., Ben Ayed, I., McCaffrey, L., Granger, E.: Tcam: Temporal class activation maps for object localization in weakly-labeled unconstrained videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 137–146 (2023)
3. Bhakat, S., Ramakrishnan, G.: Anomaly detection in surveillance videos. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data. pp. 252–255 (2019)
4. Cai, R., Zhang, H., Liu, W., Gao, S., Hao, Z.: Appearance-motion memory consistency network for video anomaly detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 938–946 (2021)
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
6. Chattopadhyay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV). pp. 839–847. IEEE (2018)
7. Cheng, F., Zheng, H., Liu, Z.: From coarse to fine: Hierarchical multi-scale temporal information modeling via sub-group convolution for video action recognition. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2021)
8. El-Tahan, K., Torki, M.: Semi-supervised anomaly detection for weakly-annotated videos. In: VISIGRAPP (5: VISAPP). pp. 871–878 (2022)
9. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)
10. Feng, J.C., Hong, F.T., Zheng, W.S.: Mist: Multiple instance self-training framework for video anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14009–14018 (2021)
11. Georgescu, M.I., Barbalau, A., Ionescu, R.T., Khan, F.S., Popescu, M., Shah, M.: Anomaly detection in video via self-supervised and multi-task learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12742–12752 (2021)
12. Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., Wei, Y.: Layercam: Exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.* **30**, 5875–5888 (2021)
13. Li, S., Liu, F., Jiao, L.: Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1395–1403 (2022)



14. Liu, H., Liu, Y., Chen, Y., Yuan, C., Li, B., Hu, W.: Transkeleton: Hierarchical spatial-temporal transformer for skeleton-based action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2023)
15. Liu, K., Ma, H.: Exploring background-bias for anomaly detection in surveillance videos. In: *Proceedings of the 27th ACM International Conference on Multimedia*. pp. 1490–1499 (2019)
16. Liu, Q., Ramanathan, V., Mahajan, D., Yuille, A., Yang, Z.: Weakly supervised instance segmentation for videos with temporal mask consistency. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13968–13978 (2021)
17. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection—a new baseline. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6536–6545 (2018)
18. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. pp. 618–626 (2017)
19. Shen, L., Li, Z., Kwok, J.: Timeseries anomaly detection using temporal hierarchical one-class network. *Adv. Neural. Inf. Process. Syst.* **33**, 13016–13026 (2020)
20. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 6479–6488 (2018)
21. Sun, L., Chen, Y., Luo, W., Wu, H., Zhang, C.: Discriminative clip mining for video anomaly detection. In: *2020 IEEE International Conference on Image Processing (ICIP)*. pp. 2121–2125. IEEE (2020)
22. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 4975–4986 (2021)
23. Wan, B., Fang, Y., Xia, X., Mei, J.: Weakly supervised video anomaly detection via center-guided discriminative learning. In: *2020 IEEE international conference on multimedia and expo (ICME)*. pp. 1–6. IEEE (2020)
24. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-cam: Score-weighted visual explanations for convolutional neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. pp. 24–25 (2020)
25. Yan, C., Zhang, S., Liu, Y., Pang, G., Wang, W.: Feature prediction diffusion model for video anomaly detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 5527–5537 (2023)
26. Zaheer, M.Z., Lee, J.h., Astrid, M., Lee, S.I.: Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 14183–14193 (2020)
27. Zhang, C., Li, G., Qi, Y., Wang, S., Qing, L., Huang, Q., Yang, M.H.: Exploiting completeness and uncertainty of pseudo labels for weakly supervised video anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16271–16280 (2023)
28. Zhang, J., Qing, L., Miao, J.: Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. In: *2019 IEEE International Conference on Image Processing (ICIP)*. pp. 4030–4034. IEEE (2019)
29. Zhong, J.X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 1237–1246 (2019)
30. Zhu, Y., Newsam, S.: Motion-aware feature for improved video anomaly detection. *arXiv preprint arXiv:1907.10211* (2019)



# EchoGCN: An Echo Graph Convolutional Network for Skeleton-Based Action Recognition

Weiwen Qian, Qian Huang<sup>(✉)</sup>, Chang Li, Zhongqi Chen, and Yingchi Mao

College of Computer Science and Software Engineering,  
Hohai University, Nanjing, China

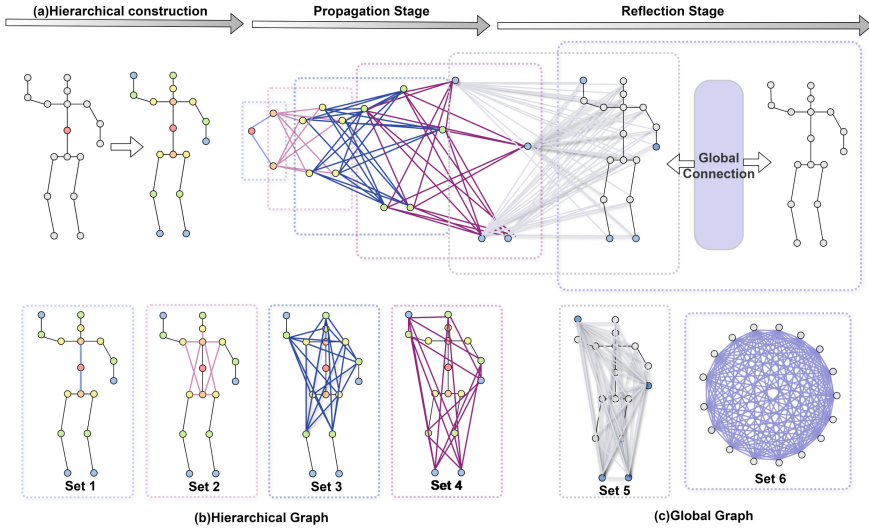
{qianweiwen, huangqian, lichang, chenzhongqi, yingchimaoy}@hhu.edu.cn

**Abstract.** Graph Convolutional Networks (GCNs) have attracted considerable attention in the realm of human action recognition. However, conventional GCNs-based methods typically struggle to construct adjacency matrices that capture diverse semantics, thus leading to performance limitations. To tackle this issue, we propose the Echo Graph as a set of adjacency matrices, which includes both hierarchical and global graphs. Specifically, our hierarchical graph exploits the hierarchical information based on the selected central broadcast nodes, aggregating joints dispersed across considerable physical distances into a unified semantic space. The global graph we construct transcends the limitations imposed by physically defined topological structures, delving into comprehensive information exchange among nodes. Additionally, we propose node activation and hierarchical activation model. These activations aim to prominently highlight crucial nodes and edges for specific samples. Finally, we incorporate Margin ReLU distillation to improve computational efficiency and design a four-stream integration using only the joint and bone data streams from two central broadcast nodes. Based on the aforementioned components, we propose EchoGCN, capable of extracting representative skeleton features. Experimental results on three datasets (NTU-RGB+D 60, NTU-RGB+D 120, and NW-UCLA) demonstrate that our model achieves state-of-the-art performance.

**Keywords:** Action recognition · skeleton · graph convolutional network · knowledge distillation

## 1 Introduction

In recent years, human action recognition has found widespread applications in various fields such as video surveillance[21], robotics[32] and human-computer interaction[12]. Thanks to the development of depth sensors and its remarkable adaptability to environmental changes, skeleton-based action recognition stands out among other human action recognition techniques such as RGB [13], depth[11] and infrared images [1].



**Fig. 1.** (a) Based on the central broadcast node (marked in red), hierarchical information is obtained through expansion. (b) Constructing hierarchical graph in Propagation Stage based on hierarchical information. (c) Connecting the nodes in the last layer of hierarchical information to all nodes to construct the global graph in Reflection Stage.

With the rise of deep learning, methods based on Recurrent Neural Networks (RNNs) [7] and Convolutional Neural Networks (CNNs) [6] have become mainstream. However, RNNs-based approaches tend to focus primarily on skeleton temporal information, while CNNs-based methods emphasize skeleton local information. Neither of them effectively combines these two aspects, leading to challenges in exploring topological structure information, where the topological structure mainly represents the connections between human skeleton joints, typically depicted as a graph structure. Graph Convolutional Networks (GCNs) effectively capture both temporal dynamics and local spatial characteristics inherent in skeleton sequences, and excel in processing graphs within non-Euclidean spaces, making them an ideal tool for extracting skeleton topology features.

Yan et al. proposed ST-GCN, which represents joints in skeleton sequences as nodes and physical connections as edges. They input the constructed graph into spatial and temporal GCN modules, successfully exploring the potential of skeleton data in modeling spatio-temporal relationships, laying the foundation for future research like [18, 25]. However, these models construct the topology structure using physical connection, resulting in difficulties capturing associations with distant joints. Previous works [3–5, 20] emphasized the optimization of the topological structure as a crucial factor for enhancing model performance. However, they did not effectively consider both hierarchical and global relationships among nodes. This results in the issue of a semantically singular adjacency matrix construction, leading to limitations in recognition performance. In this

paper, we present the propagation stage and the reflection stage to construct the new topology structure Echo Graph to address this issue. Firstly, we select a central broadcast node. As shown in Fig. 1(a), we designate the belly as the central broadcast node and then expand layer by layer based on the physical connections to construct hierarchical information. As shown in Fig. 1(b), based on hierarchical information, the propagation stage connect the nodes from the previous layer to the nodes from the subsequent layer and build a hierarchical graph. To explore global relationships fully, we design the reflection stage, as shown in Fig. 1(c). We establish a global graph by connecting the nodes of the last layer from hierarchical information to all nodes, facilitating information sharing across all hierarchical layers. In this procedure, expansion starts from a specific node, travels hierarchically to each node, and then reflects back to all nodes in the skeleton, resembling an echoing transmission and feedback. Afterwards, we refer to this structure as the Echo Graph.

Different action behaviors may involve distinct key nodes and edges, i.e., the contributions of nodes and edges to actions vary. Thus, we introduce the node and hierarchical activation methods. These methods help identify nodes and edges that significantly contribute to action recognition in specific samples, assigning distinct weights to them. In node activation, we utilize k-nearest neighbor (k-NN) operations to create a local graph in the feature space for activating similar nodes. In hierarchical activation, to address scaling biases arising from varying contributions, the process begins with max pooling in the temporal dimension of feature to extract representative features. Then, it traverses each layer of the Echo Graph for spatial average pooling to obtain hierarchical features. In hierarchical activation, features from each hierarchy are treated as nodes, and the determination of activated features is based on similarity.

However, most current GCN-based models have excessively high computational costs. For example, ST-GCN [31] has FLOPs of 16.3G, and 2S-GCN [25] even reaches 37.3G. Thus, we need a method to further improve efficiency. We incorporate Margin ReLU Distillation [8] into EchoGCN to reduce the model’s size without sacrificing recognition accuracy. Specifically, we respectively select three features from the multiple blocks in teacher and student model before the ReLU activation, and a partial  $L_2$  distance function is employed to eliminate redundant information. Moreover, we depart from the traditional four-stream integration, i.e., joint, bone, joint motion, and bone motion stream. Instead, we turn to utilize only joint and bone stream formed based on selected two central broadcast nodes.

The main contributions of this paper can be summarized as follows:

- We designed an Echo Graph, capable of capturing hierarchical and global relationships simultaneously, thereby optimizing the topological structure.
- We proposed two activation methods, aiming to activate edges and nodes crucial for recognizing specific samples.
- We incorporated the Margin ReLU distillation for model compression. Moreover, we adopted a four-way integration utilizing only joint and bone stream.

- Based on the innovation mentioned above, we established EchoGCN. Extensive experiments demonstrated the superiority of the EchoGCN, consistently achieving optimal results over the three datasets: NTU RGB+D 60[24], NTU-RGB+D 120[19], and Northwestern-UCLA [28].

## 2 Related Work

### 2.1 GCNs-based Action Recognition Methods

ST-GCN [31] first introduced GCNs into the skeleton-based action recognition, ST-GCN simultaneously constructs the spatial configuration and temporal dynamics of data. Li et al. [18] proposed AS-GCN, which integrates multi-scale modeling and generates additional human body poses for recognition tasks. Shi et al. [25] proposed 2S-GCN, which combines bone information with traditional joint information. However, these models only consider direct joint connections, ignoring distant joint relationships, thereby limiting the model’s performance. MS-G3D [20] integrated a decoupled multi-scale aggregation scheme, effectively eliminating redundant dependency relationships between different neighborhoods. However, the refined topology still relies on physical connections, with single semantics and limited feature extraction flexibility due to a uniform topology across channels. To address the above issues, Chen et al. [3] introduced CTR-GCN, which simultaneously learns a shared topology structure and channel-specific correlation matrices to obtain channel-level topology in a refined manner. However, this method overlooks the physical priors of the human skeleton, resulting in excessive flexibility during the learning process of the aforementioned topology. Shift-GCN [4] introduced spatial and temporal shift graph convolution to replace traditional spatial temporal graph convolution, reducing computational costs and adaptively adjusting the receptive field. Shift-GCN++ [5] extended Shift-GCN to achieve a more lightweight model. Although the authors designed local and global spatial convolution schemes in Shift-GCN and Shift-GCN++, this model cannot simultaneously use local and global spatial convolutions. Song et al. [27] proposed EfficientGCN-B4, which merges multiple input branches in the initial stage, effectively reducing redundant parameters. DD-GCN [16] optimize graph convolution kernel weight sharing through an activity partition strategy, and introduce a spatiotemporal synchronization encoder for embedding synchronized semantics. Li et al. [17] introduced SaPR-GCN, which divides the skeleton into body parts and employs a dynamic scale-aware mechanism to extract context-dependent multi-scale features. These methods incorporate rich semantic relationships. EfficientGCN considers joint positions, motion velocity, and skeletal features, while DD-GCN and SaPR-GCN refine topology at the part level. However, these models are limited to local relationships and overlook global relationships between joints.

To address these shortcomings, we propose a new model that captures multiple semantics while preserving skeletal structure. It considers both hierarchical and global relationships using multiple broadcast points and two topology construction methods.

## 2.2 Knowledge Distillation Methods

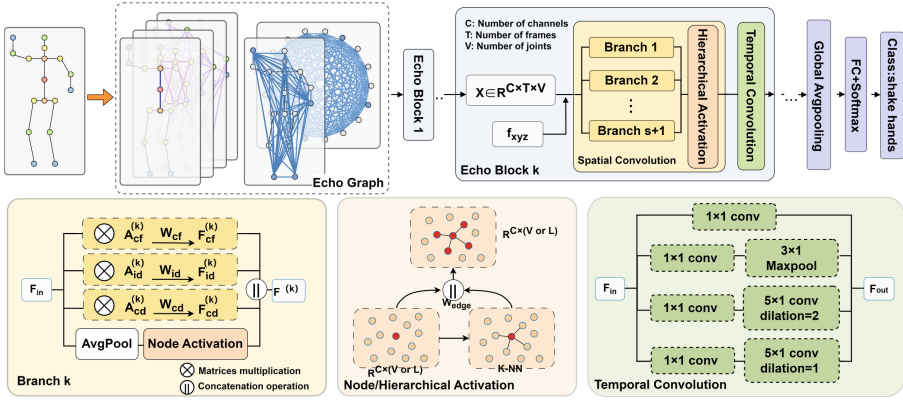
Hinton et al. [10] pioneered knowledge distillation, transferring knowledge from a teacher model to a student model through the Softmax output. Since high-performance teacher models produce outputs nearly identical to true labels, teaching these to student models limits performance. Thus, researchers are shifting from output distillation to feature distillation. FitNets [23] promoting the emulation of hidden features from the teacher model by the student model. However, this method is generally effective for higher-level hidden layers but not for lower-level hidden layers. Zagoruyko et al. [33] proposed AT, which transfers attention maps from a powerful teacher network to a smaller student network. Kim et al. [14] suggested FT, extracting teacher network factors in an unsupervised manner and transferring them to the student model. However, these methods compress the feature of the teacher model, leading to information loss. AB [9] proposed a method to transfer activation boundaries of hidden neurons, enhancing performance in classification tasks. While it partially mitigates information loss, feature distortion in the teacher model remains a challenge. Heo et al. [8] introduced the Margin ReLU Distillation, adjusting distillation locations to before the ReLU activation layer and using an  $L_2$  distance function to filter unnecessary information. This method preserves valuable information, eliminates unfavorable aspects, and prevents feature distortion. Thus, to enhance performance and reduce model parameter size, we leverage the effectiveness of Margin ReLU distillation in our approach.

## 3 Methodology

The overall framework of our proposed EchoGCN is illustrated in the Fig. 2. Firstly, we construct the Echo Graph based on the original skeleton graph. Subsequently, we propagate features into a network comprising multiple Echo Blocks. Each Echo Block consists of a spatial convolution and a temporal convolution. The spatial convolution includes  $s+1$  branches (yellow), with each branch comprising 3 GCNs and a node activation (orange). After concatenating the results of  $s+1$  branches, they are transmitted to the hierarchical activation (orange), highlighting important hierarchical sets. Finally, it goes through a temporal convolution (green). The process concludes with a global average pooling, an FC layer, and Softmax, resulting in the output classification. In the following section, we will provide a detailed explanation of each component.

### 3.1 Echo Graph Constructing

**Propagation Stage** Traditional action recognition using GCNs face challenges that make it difficult to effectively capture relationships between distant joints. Consider clapping as an example: accurately understanding the relationship between two hands that are physically distant is crucial for the proper recognition of clapping actions in such scenarios. To solve this problem, we construct



**Fig. 2.** The overall architecture of the model.

topological structure through hierarchical information. Firstly, a central broadcast node is set, and layer-wise expansion is carried out in a centrifugal manner based on the physical connections. As illustrated in Fig. 1(a), the belly is designated as the central broadcast node and is individually stored in a node set (depicted in red). Next, a layer of expansion is conducted outward from the belly, spreading the chest and hip to be stored together in the same node set (depicted in orange). Subsequently, expansion layers extend from the chest and hip, involving joints of the neck, left and right shoulders, and left and right hip, and storing them in the same semantic space, effectively addressing the issue of long-range dependencies.

Once all nodes have undergone expansion,  $s$  sets of hierarchical informations  $S$  are formed, represented as  $S_k$  for the  $k$ -th set. As shown in Fig. 1(b), all nodes in  $S_k$  are bidirectionally connected to all nodes in  $S_{(k+1)}$ , resulting in the formation of edge sets  $E_{cp}(S_k \rightarrow S_{(k+1)})$  and  $E_{cf}(S_k \leftarrow S_{(k+1)})$ , representing centrifugal and centripetal edges, respectively. Additionally, a self-connecting edge  $E_{id}(S_k \cup S_{(k+1)})$  is formed for the node set  $S_k \cup S_{(k+1)}$ . Thus,  $s - 1$  hierarchical edge sets are obtained:

$$E_{Hi} = \left[ \left\{ \begin{matrix} E_{id}(S_1 \cup S_2) \\ E_{cp}(S_1 \rightarrow S_2) \\ E_{cf}(S_1 \leftarrow S_2) \end{matrix} \right\}, \dots, \left\{ \begin{matrix} E_{id}(S_{(s-1)} \cup S_s) \\ E_{cp}(S_{(s-1)} \rightarrow S_s) \\ E_{cf}(S_{(s-1)} \leftarrow S_s) \end{matrix} \right\} \right] \quad (1)$$

**Reflection Stage** However, the hierarchical graph unable to fully explore the global relationships between nodes. Therefore, we introduce the reflection stage, as shown in Fig. 1(c). We connect the last layer’s nodes (i.e,  $S_s$ ) from the hierarchical information to full nodes, achieving information propagation on a global scale. The specific design is as follows.

We design full node sets  $S_{(s+1)}$  and  $S_{(s+2)}$ , where  $S_{(s+1)} = S_{(s+2)} = \{v_i | i = 1, \dots, N\}$ , and represent the global edge sets as below:

$$E_{\text{Go}} = \left[ \left[ \begin{array}{l} E_{\text{id}}(S_s \cup S_{(s+1)}) \\ E_{\text{cp}}(S_s \rightarrow S_{(s+1)}) \\ E_{\text{cf}}(S_s \leftarrow S_{(s+1)}) \end{array} \right], \left[ \begin{array}{l} E_{\text{id}}(S_{(s+1)} \cup S_{(s+2)}) \\ E_{\text{cp}}(S_{(s+1)} \rightarrow S_{(s+2)}) \\ E_{\text{cf}}(S_{(s+1)} \leftarrow S_{(s+2)}) \end{array} \right] \right] \quad (2)$$

We denote the final edge sets  $E = [E_{\text{Hi}}, E_{\text{Go}}]$ . Through node set  $V$  and edge set  $E$ , we construct the matrix  $A_{\text{HiGo}} \in \mathbb{R}^{(s+1) \times 3 \times N \times N}$  includes both hierarchical and global graph.

### 3.2 Spatial Convolution

**Convolution Operation** The first part of the spatial graph convolution includes  $s - 1$  hierarchical branches and two global branches. Each branch consists of four parallel branches: three GCN operations and one node activation. The specific design is illustrated in the left part of Fig. 2 in yellow.

Firstly, beside the first block of the network, we concatenate the features  $F$  from the output of the previous block with the three-dimensional x-y-z coordinates  $f_{xyz}$  of the skeleton. This is carried out to enhance the precision of comprehending geometric concepts in the spatial domain, especially for high-level features. The processed features are denoted as  $F_{\text{in}} = \phi(f_{xyz} || F)$ , where  $||$  represents the concatenation operation along the channel dimension.  $\phi(\cdot)$  denotes a linear transformation to reduce the dimensionality. Next, we perform GCN operations on  $A_{\text{HiGo}}$  for three subsets and concatenate the output values along the channel dimension:

$$F^{(k)} = \parallel_{p \in P} \{ \hat{A}_{\text{HiGo}}^{(k)} F_{\text{in}} W_p^{(k)} \} \quad (3)$$

where  $F^{(k)}$  denotes the concatenation from the three GCN operations in the k-th layer branch. Define  $P = \{root, cp, cf\}$ , where *root*, *cp*, and *cf* denote subsets of self, centrifugal, and centripetal nodes, respectively.  $\hat{A}_{\text{HiGo}}$  is the matrix obtained by normalizing  $A_{\text{HiGo}}$ , and  $W_p^{(k)}$  denotes the weights of the  $1 \times 1$  convolution for the p-th partition of the k-th layer.

**Node Activation** Although we defined hierarchical and global edge sets in the Propagation and Reflection Stages, the model still cannot accurately capture the relationships that reflect the similarity between all nodes in the feature space.

We know that different data samples may involve different nodes, in the node activation, we form a neighborhood graph in the feature space using k-NN operations to extract graphical features. Specifically, we first perform average pooling on  $F_{\text{in}}$  along the temporal dimension to obtain the overall characteristics of the specific action and simplify computational complexity. Then, we use k-NN operations to aggregate the features of neighborhood edges in the feature space to form a neighborhood graph. Furthermore, we aggregate the features of the node itself to accurately reflect its characteristics:



$$F_{\text{fe}}^{(k)} = \omega_1(\text{kNN}(\text{Avg}_T(F_{\text{in}})) \parallel \text{Avg}_T(F_{\text{in}})) \quad (4)$$

where  $\text{Avg}_T(\cdot)$  denotes the average pooling along the temporal dimension, and  $\omega_1(\cdot)$  denotes the aggregation operation with  $W \in \mathbb{R}^{C \times 2C}$ , where  $C$  is the channel dimension of  $\text{Avg}_T(F_{\text{in}})$ . We obtain the final feature via summation from the results of equations (3) and (4):

$$F = \sum_{k=1}^{(s+1)} [F^{(k)} \parallel F_{\text{fe}}^{(k)}] \quad (5)$$

**Hierarchical Activation** Similarly, different data samples may involve different edges. Therefore, it is essential to identify the edges contributing differently to action recognition in specific samples and assign corresponding weights to them. Thus, we propose an hierarchical activation to implement this concept.

Specifically, we first perform max pooling along the temporal dimension to obtain a representative feature  $F_{\text{max}} \in \mathbb{R}^{C \times (l+2) \times V}$ . Then, we additionally introduce a node extraction operation to address scaling bias, since the number of edges connected to each joint is different. We traverse the node set constructed in Section 3.1 hierarchically, extracting only the features of nodes exist in each layer from  $F_{\text{max}}$ . Then, we perform spatial average pooling on the features within the node set to extract hierarchical features:

$$F_{\text{level}}^{(k)} = \frac{1}{N_k + N_{(k+1)}} \sum_{v \in S_k \cup S_{(k+1)}} F_{\text{max}}^{(k)}(v) \quad (6)$$

Finally, similar to the node activation, we calculate the similarity of hierarchical features to determine which features should be highlighted:

$$M = \sigma(\omega_2(\text{kNN}(\parallel_{k \in (l+2)} F_{\text{level}}^{(k)}) \parallel (\parallel_{k \in (l+2)} F_{\text{level}}^{(k)}))) \quad (7)$$

where  $\sigma(\cdot)$  represents the sigmoid function. We then perform element-wise multiplication between the obtained  $M$  and the features  $F$  outputted in Equation (5). Finally, we get the output in the spatial convolution block.

### 3.3 Temporal Convolution

Our temporal convolution module, inspired by [3], consists of four branches. Each branch incorporates a  $1 \times 1$  convolution to reduce channel size. The first three branches additionally feature two temporal convolutions with kernel sizes of 5, dilation=[1,2], and a max pooling with a kernel size of 3. The final output is obtained by concatenating the results from the four branches.

### 3.4 Margin ReLU Distillation

We utilize the Margin ReLU Distillation to compress our proposed model. We assume that the scale of the teacher model network is (L,C,T), where L, C, T represent the number of layers, channels, and frames of the model, respectively. To simplify the network, we set the scale of the student model to (2/3 L, 1/2 C, 1T). We selected three features from each model that had not been activated by the ReLU function for distillation. We utilize the positive responses from the teacher network for transmission and adjustment of the student network. Conversely, for the negative responses in the teacher network, the student network generates a value less than the target value to ensure the same activation state. Initially, we use this function to transform the teacher features:

$$\sigma_{m_c}(F_t) = \max(F_t, m_c) \quad (8)$$

where  $F_t$  denotes the teacher feature, and  $m_c$  is less than 0, derived from the parameters of the preceding batch normalization layer.  $\sigma_{m_c}(\cdot)$  denotes the Margin ReLU function.

To avoid information loss, we use a  $1 \times 1$  convolution and batch normalization layer operation  $r(\cdot)$  to change the features of the student. We expand the dimensions of the student model features  $F_s$  to  $r(F_s)$ . Finally, Margin ReLU Distillation adopts a partial  $L_2$  loss  $d_p$ , and we represent the Margin ReLU Distillation loss as:

$$L_{\text{distill}} = d_p(\sigma_{m_c}(F_t), (r(F_s))) \quad (9)$$

$$d_p(T, S) = \sum_i^N \begin{cases} 0 & \text{if } S_i \leq T_i \leq 0 \\ (T_i - S_i)^2 & \text{otherwise} \end{cases} \quad (10)$$

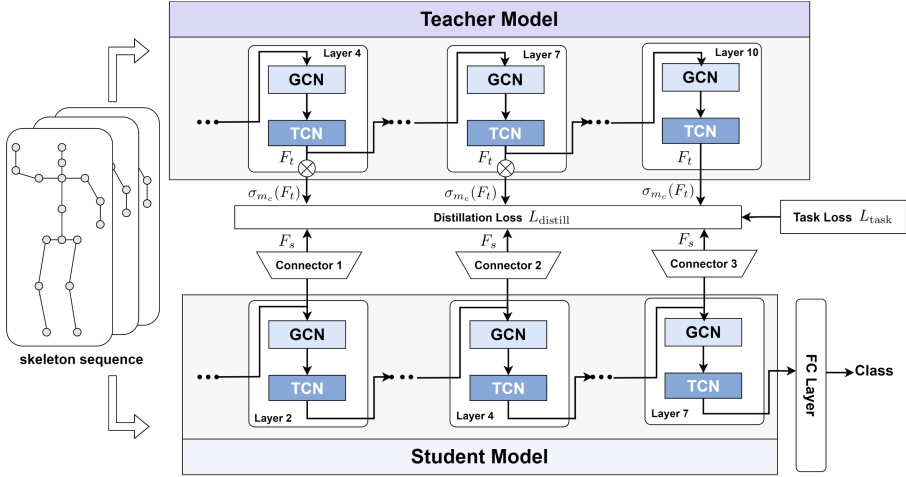
where  $T_i$  and  $S_i$  are the  $i$ -th components of the tensors. The final loss function is the sum of the distillation loss, weighted by manually defined coefficients  $\alpha L_{\text{distill}}$ , and the task loss  $L_{\text{task}}$ .

### 3.5 Four-Way Integration

In recent years, human action recognition commonly utilize a four-stream integration: joint, bone, joint motion, and bone motion streams. However, to simplify the model, we discard the motion streams and use only the joint and bone streams from two selected central broadcast nodes, assigning the same contribution coefficients to both. In experiments, EchoGCN outperforms state-of-the-art methods without using motion streams or manually setting ensemble coefficients.

### 3.6 Network Architecture

The architecture of the distillation network is illustrated in Figure 3. Our teacher model consists of 10 Echo blocks, with output channel dimensions for each block



**Fig. 3.** The distillation network architecture of the model. We transmit the skeleton sequences to both the teacher and student models, and extract features before ReLU activation from the teacher model at layers 4, 7, and 10, and from the student model at layers 2, 4, and 7 for distillation.  $\otimes$  denotes  $\sigma_{m_c}(\cdot)$ , and Connector  $x$  ( $x = 1, 2, 3$ ) denotes  $r(\cdot)$  in Section 3.4. Finally, we perform classification on the student model.

being: 64-64-64-64-128-128-128-256-256-256. For the student model, there are a total of 7 Echo blocks, with output channel dimensions for each block being: 32-32-64-64-128-128-128. Each block also includes a residual connection. We extract features from the teacher model at layers 4, 7, and 10 before ReLU activation, and from the student model at layers 2, 4, and 7 before ReLU activation for distillation. We perform sample classification solely on the student model. After passing through the 7 Echo blocks of the student model, we classify action samples using a global average pooling layer and a softmax function.

## 4 Experiment

### 4.1 Datasets

**NTU-RGB+D 60:** NTU-RGB+D 60[24] is one of the most widely used 3D skeleton datasets in the field of human action recognition. Captured by Kinect V2 at 30 fps, it includes 56,880 skeleton action samples from 40 subjects across 60 categories and 3 perspectives. We follow two recommended metrics: Cross-Subject (X-Sub) and Cross-View (X-View).

**NTU-RGB+D 120:** NTU-RGB+D 120[19] extends the NTU-RGB+D 60 dataset with samples from 106 subjects, totaling 114,480 skeleton action samples across 120 categories from 3 perspectives. We evaluate using two recommended metrics: Cross-Subject (X-Sub) and Cross-Setup (X-Set).

**Northwestern-UCLA:** Northwestern-UCLA[28] includes 1494 video clips covering 3 perspectives, 10 categories, and each action category is performed

by 10 different subjects. We use the same evaluation method as [28]: using data samples from the first two camera perspectives as training data and data samples from the other camera perspective as testing data.

**Experimental Setup:** Our experiments are based on PyTorch and were conducted on two RTX 3090 GPUs. We trained for 90 epochs with a warm-up strategy for the initial 5 epochs. The SGD optimizer was used with a momentum of 0.9 and weight decay of 0.0004. Learning rate decay followed a cosine annealing schedule, ranging from a maximum of 0.1 to a minimum of 0.0001. For NTU-RGB+D, the batch size was 64 for the non-distilled model and 48 for the distilled model. For Northwestern-UCLA, the batch size remained 16 for both non-distilled and distilled models.

## 4.2 Comparison with State-of-the-Art Methods

We conducted experiments on three datasets: NTU-RGB+D 60, 120, and Northwestern-UCLA, comparing our proposed EchoGCN and EchoGCN-distill with state-of-the-art networks, as shown in Tables 1 and 2. For NTU-RGB+D 60, we report the Top-1 and Top-5 accuracies for X-sub and X-view. For NTU-RGB+D 120, we show the Top-1 and Top-5 accuracies for X-sub and X-set. Additionally, we provide the Floating-Point Operations (FLOPs) and model parameters (Para.) for these datasets. In the UCLA dataset, we also report the Top-1 and Top-5 accuracies, along with FLOPs and model parameters.

**Table 1.** Comparisons of the top-1 and top-5 accuracy (%) , FLOPs and model parameters with the state-of-the-art methods on the NTU RGB + D 60 and 120 datasets.

Methods	NTU 60				NTU 120				FLOPs	Para.
	X-Sub		X-View		X-Sub		X-Set			
	Top-1	Top-5*	Top-1	Top-5*	Top-1	Top-5*	Top-1	Top-5*		
ST-GCN(1s)[31]	81.5	96.9	88.3	98.5	70.7	95.0	73.2	96.6	16.32G*	3.10M*
AS-GCN(1s)[18]	86.8	97.3	94.2	99.1	78.3	96.9	79.8	97.0	26.76G*	9.50M*
2S-GCN(2s)[25]	88.5	98.5	95.2	99.2	82.5	97.8	84.2	97.8	37.32G*	6.94M*
Shift-GCN(4s)[4]	90.7	98.8	96.5	99.4	85.9	97.6	87.6	98.1	10.0G	2.76M
Shift-GCN++(4s)[5]	90.5	98.7	96.3	99.3	85.6	97.5	87.2	98.0	1.70G	1.80M
MS-G3D(2s)[20]	91.5	98.6	96.2	99.3	86.9	98.1	88.4	98.3	10.44G*	6.44M*
CTR-GCN(4s)[3]	92.4	99.1	96.8	99.3	88.9	98.6	90.6	98.5	7.88G*	5.84M*
EfficientGCN(3s)[27]	91.7	98.9	95.7	99.2	88.3	98.3	89.1	98.3	8.36G	1.10M
DD-GCN(2s)[16]	92.6	-	96.9	-	88.9	-	90.2	-	-	-
HGCT(4s)[2]	92.2	-	96.5	-	89.2	-	90.6	-	-	-
SaPR-GCN(4s)[17]	92.4	-	96.4	-	88.7	-	90.3	-	6.60G	8.28M
STHG-DAN(3s)[30]	91.2	-	96.5	-	88.7	-	89.8	-	5.18G	2.65M
ACE-ens(2s)[22]	91.6	-	96.3	-	88.2	-	89.2	-	78.0G	5.80M
EchoGCN(2s)	<b>92.4</b>	99.2	96.5	99.4	89.1	98.7	<b>90.3</b>	<b>98.7</b>	4.2G	4.72M
EchoGCN(4s)	92.7	99.2	<b>96.9</b>	<b>99.5</b>	89.5	98.8	90.7	98.7	8.4G	9.44M
EchoGCN-distill(2s)	92.3	<b>99.2</b>	<b>96.5</b>	<b>99.4</b>	<b>89.2</b>	<b>98.7</b>	90.2	98.6	0.7G	1.32M
EchoGCN-distill(4s)	<b>92.8</b>	<b>99.3</b>	96.7	99.5	<b>89.7</b>	<b>98.8</b>	<b>90.9</b>	<b>99.0</b>	1.4G	2.64M

<sup>1</sup> Those marked with \* are the results from the corresponding methods we reproduced

**Table 2.** Comparisons of the top-1 and top-5 accuracy (%) , FLOPs and model parameters with the state-of-the-art methods on the Northwestern-UCLA dataset.

Methods	UCLA		FLOPs	Para.
	Top-1	Top-5*		
TS-LSTM(4s)[15]	89.2	99.1	-	-
AGC-LSTM(2s)[26]	93.3	99.2	-	-
Shift-GCN(4s)[4]	94.6	99.3	0.70G	1.28M
Shift-GCN++(4s)[5]	95.0	99.3	0.11G	0.44M
CTR-GCN(4s)[3]	96.5	99.4	2.32G*	5.64M*
Graph2Net(2s)[29]	95.3	99.3	0.64G*	1.62M*
SaPR-GCN(4s)[17]	96.6	-	1.31G	2.06M
EchoGCN(2s)	95.9	99.4	2.38G	3.96M
EchoGCN(4s)	96.3	99.5	4.76G	7.92M
EchoGCN-distill(2s)	<b>96.2</b>	<b>99.4</b>	0.38G	1.10M
EchoGCN-distill(4s)	<b>96.7</b>	<b>99.5</b>	0.76G	2.20M

<sup>1</sup> Those marked with \* are the results from the corresponding methods we reproduced

Overall, our EchoGCN and EchoGCN-distill outperform existing state-of-the-art models across all datasets with fewer FLOPs and parameters. Specifically, EchoGCN-distill achieves slightly better results than EchoGCN while significantly reducing FLOPs and parameters, and the 4-stream modality outperforms the 2-stream modality, validating the effectiveness of our integrated Margin ReLU distillation and four-stream ensemble approach. In the NTU-RGB+D 60 and 120 datasets, ShiftGCN, ShiftGCN++, MS-G3D, and CTR-GCN perform better than ST-GCN, AS-GCN, and 2S-GCN. This highlights the limitations of models that only consider single-hop physical connections in skeletal topology while neglecting distant joint relationships. However, ShiftGCN, ShiftGCN++, and MS-G3D fail to balance local and global spatial convolutions or broader semantic context, while CTR-GCN’s lack of physical priors makes its topology learning overly flexible. Consequently, these models underperform compared to EfficientGCN, DDGCN, SaPR-GCN, STHG-DAN and ACE-ens, which have more thoughtfully designed topologies. Additionally, HGCT achieves commendable results due to its Transformer-based design. In the UCLA dataset, TS-LSTM and AGC-LSTM underperform all GCN-based models, highlighting that LSTM is less effective for graph-structured data like skeletal structures. The performance of other models in the UCLA dataset is similar to their performance in the NTU datasets, so we omit further details for brevity.

In conclusion, EchoGCN and EchoGCN-distill consider both hierarchical and global relationships, consistently outperform the aforementioned models, demonstrating their superiority. For detailed results on the effectiveness of each component, please refer to the subsequent ablation study section.

**Influence of Hierarchical and Global Graphs** We conducted an evaluation of the hierarchical graph (Hira) and the global graph (Gb) on NTU-RGB+D 120. These two graphs construct the hierarchical and global relationships between nodes starting from a broadcast node, enriching the semantics of the topological structure. In this set of experiments, we established four test samples: EchoGCN (our proposed model without the distillation component), w/o Hira (a model based on EchoGCN with the Hira removed), w/o Gb (a model based on EchoGCN with the Gb removed), and w/o Hira+Gb (a model based on EchoGCN with both the Hira and Gb removed).

**Table 3.** Comparison (%) of Different Graph Structure Designs

Methods	X-Sub	X-View
w/o Hira	85.1	86.9
w/o Gb	85.0	86.9
w/o Hira+Gb	84.7	86.6
EchoGCN	85.2	87.0

**Table 4.** Comparison (%) of Different Activation Designs

Methods	X-Sub	X-View
w/o HA	84.8	86.7
w/o NA	85.0	86.8
w/o HA+NA	84.5	86.4
EchoGCN	85.2	87.0

**Table 5.** Comparisons of the top-1 accuracy of Xsub and Xview(%) between Distillation Models and Non-Distillation Models

Node	Stream	NTU 60		NTU 120	
		Xsub(T/S)	Xview(T/S)	Xsub(T/S)	Xset(T/S)
Hip	Joint	90.6/90.7	95.2/95.4	85.2/86.0	87.0/87.5
	Bone	90.5/90.9	94.9/95.2	86.6/87.3	88.2/88.6
Belly	Joint	90.4/90.9	95.6/95.6	85.1/85.8	86.7/87.1
	Bone	90.6/91.0	94.8/95.2	86.0/86.8	88.3/88.7

### 4.3 Ablation Study

As shown in Table 3, EchoGCN performs significantly better than EchoGCN w/o Hira+Gb, demonstrating the effectiveness of our proposed Echo Graph (i.e., the hierarchical and global graphs). Additionally, EchoGCN w/o Hira and EchoGCN w/o Gb both outperform EchoGCN w/o Hira+Gb, indicating that both the hierarchical graph and the global graph can effectively enhance the model’s performance. Finally, by comparing the results of EchoGCN w/o Hira and EchoGCN w/o Gb, we can conclude that the global graph provides a more significant improvement to the model’s performance.

**Role of Activation** We introduce two activation, namely Node Activation (NA) and Hierarchical Activation (HA). These two components help identify nodes and edges that significantly contribute to action recognition in specific samples, and assign distinct weights to them. In this section, we validate their effects on NTU-RGB+D 120. We established four test samples: EchoGCN (our baseline model), w/o HA (a model based on EchoGCN with HA removed), w/o NA (a model based on EchoGCN with NA removed), and w/o HA+NA (a model based on EchoGCN with both HA and NA removed).

As shown in Table 4, EchoGCN w/o HA and EchoGCN w/o NA both outperform EchoGCN w/o HA+NA, demonstrating both hierarchical activation and node activation are effective in enhancing the model’s performance. Furthermore, EchoGCN performs significantly better than EchoGCN w/o HA+NA, proving the effectiveness and rationality of integrating both hierarchical and node activations. Additionally, the performance of EchoGCN w/o NA surpasses that of EchoGCN w/o HA, indicating node activation contributes more significantly to the improvement of the model’s performance. Although the design of hierarchical activation is reasonable, a more fine-grained node activation is needed.

**Effect of Knowledge Distillation** To tackle the challenge of model complexity, we incorporate Margin ReLU distillation into EchoGCN. The single-stream experimental results on the NTU-RGB+D60 and 120 datasets are presented in Table 5. We abbreviate “Teacher model,” i.e., EchoGCN, as “T” and “Student model,” i.e., EchoGCN-distill, as “S.” Combining the insights from Table 1 and Table 2, it is evident that EchoGCN-distill consistently outperforms EchoGCN in single-stream predictions while having fewer FLOPs and parameters. These experimental outcomes affirm the effectiveness of Margin ReLU distillation.

## 5 Conclusion

In this work, we proposed a Echo Graph convolutional model for skeleton action recognition (EchoGCN). we introduce the Echo Graph, which includes hierarchical and global graphs that simultaneously considers hierarchical and global topological structures. Additionally, we introduce node and hierarchical activation to highlight crucial nodes and edges for specific samples. We integrate the Margin ReLU Distillation for boosting the efficiency and propose a novel four-way integration only using joint and bone stream. On three datasets(NTU-RGB+D 60, NTU-RGB+D 120, and NW-UCLA), the performance of the proposed EchoGCN surpasses that of state-of-the-art methods.

**Acknowledgements.** This work is supported by the Key Research and Development Program of China (No. 2022YFC3005401), the Key Research and Development Program of China, Yunnan Province (No. 202203AA080009), the Fundamental Research Funds for the Central Universities (No. B230205027), Postgraduate Research & Practice Innovation Program of Jiangsu Province(No. KYCX23\_0753), the 14th Five-Year Plan for Educational Science of Jiangsu Province (No. D/2021/01/39), the Jiangsu

Higher Education Reform Research Project (No. 2021JSJG143) and the 2022 Undergraduate Practice Teaching Reform Research Project of Hohai University.

## References

1. Akula, A., Shah, A.K., Ghosh, R.: Deep learning approach for human action recognition in infrared images. *Cogn. Syst. Res.* **50**, 146–154 (2018)
2. Bai, R., Li, M., Meng, B., Li, F., Jiang, M., Ren, J., Sun, D.: Hierarchical graph convolutional skeleton transformer for action recognition. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 01–06 (2022)
3. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13339–13348 (2021)
4. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 180–189 (2020)
5. Cheng, K., Zhang, Y., He, X., Cheng, J., Lu, H.: Extremely lightweight skeleton-based action recognition with shiftgcn++. *IEEE Trans. Image Process.* **30**, 7333–7348 (2021)
6. Chéron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3218–3226 (2015)
7. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1110–1118 (2015)
8. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1921–1930 (2019)
9. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (2019)
10. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. *ArXiv abs/1503.02531* (2015)
11. Jalal, A., Kim, Y.H., Kim, Y.J., Kamal, S., Kim, D.: Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern Recogn.* **61**, 295–308 (2017)
12. Jiang, Y.G., Dai, Q., Liu, W., Xue, X., Ngo, C.W.: Human action recognition in unconstrained videos by explicit motion modeling. *IEEE Trans. Image Process.* **24**(11), 3781–3795 (2015)
13. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
14. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 2765–2774 (2018)
15. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 1012–1020 (2017)



16. Li, C., Huang, Q., Mao, Y.: Dd-gcn: Directed diffusion graph convolutional network for skeleton-based human action recognition. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 786–791 (2015)
17. Li, C., Mao, Y., Huang, Q., Zhu, X., Wu, J.: Scale-aware graph convolutional network with part-level refinement for skeleton-based human action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2023)
18. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3590–3598 (2019)
19. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(10), 2684–2701 (2020)
20. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 143–152 (2020)
21. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 935–942 (2009)
22. Qin, Z., Liu, Y., Ji, P., Kim, D., Wang, L., McKay, R.I., Anwar, S., Gedeon, T.: Fusing higher-order features in graph neural networks for skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* **35**(4), 4783–4797 (2024)
23. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: International Conference on Learning Representations (ICLR) (2015)
24. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1010–1019 (2016)
25. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12018–12027 (2019)
26. Si, C., Chen, W., Wang, W., Wang, L.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1227–1236 (2019)
27. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Constructing stronger and faster base-lines for skeleton-based action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(2), 1474–1488 (2023)
28. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.C.: Cross-view action modeling, learning, and recognition. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. p. 2649–2656 (2014)
29. Wu, C., Wu, X.J., Kittler, J.: Graph2net: Perceptually-enriched graph learning for skeleton-based action recognition. *IEEE Trans. Circuits Syst. Video Technol.* **32**(4), 2120–2132 (2022)
30. Wu, Z., Ma, N., Wang, C., Xu, C., Xu, G., Li, M.: Spatial-temporal hypergraph based on dual-stage attention network for multi-view data lightweight action recognition. *Pattern Recogn.* **151**, 110427 (2024)
31. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI on Artificial Intelligence (2018)

32. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. pp. 28-35 (2012)
33. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations (ICLR) (2017)



# From Category to Scenery: An End-to-End Framework for Multi-person Human-Object Interaction Recognition in Videos

Tanqiu Qiao , Ruo Chen Li , Frederick W. B. Li ,  
and Hubert P. H. Shum <sup>(✉)</sup> 

Durham University, Durham, UK

{tanqiu.qiao, ruochen.li, frederick.li, hubert.shum}@durham.ac.uk

**Abstract.** Video-based Human-Object Interaction (HOI) recognition explores the intricate dynamics between humans and objects, which are essential for a comprehensive understanding of human behavior and intentions. While previous work has made significant strides, effectively integrating geometric and visual features to model dynamic relationships between humans and objects in a graph framework remains a challenge. In this work, we propose a novel end-to-end category to scenery framework, CATS, starting by generating geometric features for various categories through graphs respectively, then fusing them with corresponding visual features. Subsequently, we construct a scenery interactive graph with these enhanced geometric-visual features as nodes to learn the relationships among human and object categories. This methodological advance facilitates a deeper, more structured comprehension of interactions, bridging category-specific insights with broad scenery dynamics. Our method demonstrates state-of-the-art performance on two pivotal HOI benchmarks, including the MPHOI-72 dataset for multi-person HOIs and the single-person HOI CAD-120 dataset.

**Keywords:** Human-object interaction · Multi-person interaction · Feature fusion

## 1 Introduction

Human-Object Interaction (HOI) recognition delves into the subtle dynamics between humans and objects, aiming to capture the breadth of their interactions from basic actions to complex activities. This field transcends mere identification to explore the depth of their interactions, from elementary actions to intricate sequences, which are essential for a comprehensive understanding of

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_17](https://doi.org/10.1007/978-3-031-78354-8_17).

human behavior and intentions [30, 35, 51]. Accurate HOI recognition is crucial across various domains, serving as a cornerstone for developing sophisticated surveillance [7, 37], enhancing video analysis techniques [25, 27, 32], and facilitating effective human-robot collaboration [31, 39].

Prior work in Human-Object Interaction (HOI) detection predominantly examines interactions within static images, offering crucial insights yet constrained by the lack of temporal dynamics [12, 13, 28]. The emergence of single-person HOI video datasets marks a significant advancement [8, 18, 19], enabling the development of models that understand spatio-temporal actions through visual cues [15, 30, 34]. A notable progression is presented by [35], which leverages geometric features informed networks for HOI recognition in videos, broadening the scope to encompass two-person HOIs with the introduction of a novel dataset.

While fusing geometric and visual features achieves remarkable performance, video-based HOI recognition still faces challenges in effectively fusing these features and learning dynamic relationships between humans and objects in a graph model. 2G-GCN [35] attempts to enrich visual data with geometric information via a graph-based network. However, merging geometric features of all humans and objects with individual visual features in a single graph leads to a critical flaw by neglecting category-specific characteristics. This fusion difficulty hampers accurate and specific HOI learning, especially in complex multi-person scenes.

Categorization simplifies learning and improves behavior discrimination by grouping similar features, enhancing model accuracy in identifying diverse interactions. In this work, we follow natural cognitive processes [3, 26] to learn HOIs from category-level feature fusion to scenery-level graph representation, facilitating a structured and comprehensive understanding. This strategy enables a more sophisticated integration of varied feature types, ensuring each level is fully leveraged for enhanced representational efficacy. We propose a novel end-to-end CATegory to Scenery framework (CATS), which initially generates geometric features via a graph for different categories, integrating them with corresponding visual features. Subsequently, a scenery interactive graph is constructed using these enriched geometric-visual features as nodes, to deeply understand the interaction dynamics among all humans and objects.

Our approach surpasses state-of-the-art performance on two HOI benchmarks, including the two-person MPHUI-72 [35] dataset and the single-person HOI CAD-120 [18] dataset. Additionally, we conduct ablation studies to evaluate the core components of our model. Our main contributions are:

- We propose an end-to-end framework CATS ranging from category-level feature fusion to scenery-level graph for multi-person HOI recognition in videos <sup>1</sup>.
- We propose a multi-category multi-modality fusion module that fuses visual features and graph-based geometric features for human and object categories, respectively.
- We propose a scenery interactive graph to learn the relationships among human and object categories via an attention-based graph.

## 2 Related Work

### 2.1 HOI Recognition in Videos

There are two setups for video-based HOI recognition, where the more challenging setup focuses on segmenting and recognizing distinct human sub-activities in videos. Deep neural networks (DNNs) and graphical models have been combined in recent works. A paradigm for integrating the effectiveness of spatio-temporal graphs with Recurrent Neural Networks (RNNs) in sequence learning is presented by Jain et al. [15]. Using learnable graph structures for videos, Qi et al. [34] expand previous graphical models in DNNs and pass messages through GPNN. For the intention of acquiring spatial relations, Dabral et al. [6] compare GCNs to Convolutional Networks and Capsule Networks. In attempting to investigate the evolution of spatio-temporal connections and identify objects in a scene [23, 43], STIGPN [43] utilizes visual-based multi-modal features and a multi-stream fusion strategy to enhance the reasoning capability of the model. Morais et al. [30] present a visual feature attention model to learn asynchronous and sparse HOI in videos. Xing et al. [44] represent the 2D or 3D spatial relation of human skeletons and object center points from the detection results in video data as a graph. Based on prior visual-only and geometric-only approaches, 2G-GCN [35] incorporates geometric features to complement visual features into the HOI recognition network through a graph network. Nevertheless, the fusion of geometric and visual features introduces certain design complexities that offer opportunities for further refinement.

Another more relaxed setup in HOI recognition aims to generate  $\langle \text{human}, \text{predicate}, \text{object} \rangle$  triplets, neglecting a more detailed analysis of specific actions and interactions. For example, in recent years, SERVO-HOI [1] presents a robust end-to-end framework adept at recognizing HOIs within in-the-wild videos, especially effective in high label-skew settings. Zeng et al. [47] introduce the Relation-Pose Transformer (RPT), a novel framework designed to intricately model the spatial and temporal dynamics between relations and poses, adept at encapsulating spatially contextualized information and the temporal evolution of relationships. Furthermore, Zhang et al. [49] explore a new task, Human-Object-Object Interaction (HOOI) detection, focusing on localizing the human and identifying their interactions within untrimmed videos as a quadruple  $\langle \text{human}, \text{interaction}, \text{object1}, \text{object2} \rangle$ . In this work, our study concentrates on the more challenging aspect of video-based HOI recognition, specifically the segmentation and recognition of distinct human sub-activities along the video timeline.

### 2.2 Graph-based HOI Analysis

Graphical models facilitate the sharing of contextual information among nodes. Qi et al. [34] introduce this concept in HOI detection, where they propose a fully-connected graph with detected instances as nodes and update node features with a message passing algorithm. Wang et al. [42] suggest that adaptation to

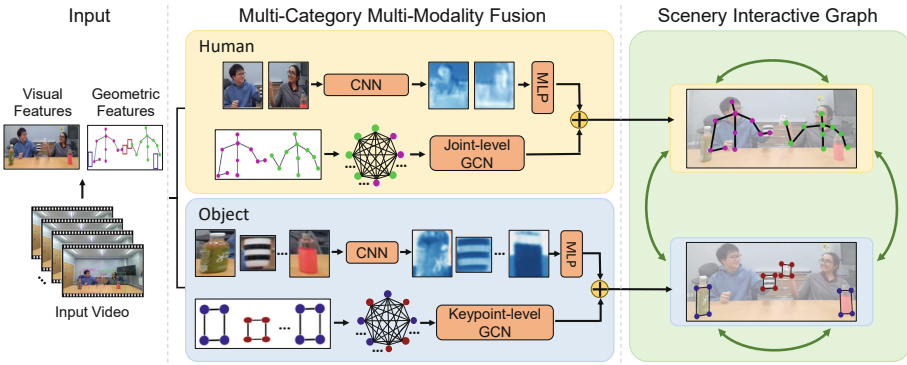
two sets of heterogeneous nodes, human and object, is essential for graph-based HOI analysis. This necessitates modelling intra-class messages differently from inter-class messages during message passing. Incorporating the heterogeneity of nodes, Gao et al. [11] create separate human-centric and object-centric graphs for HOI detection by treating human-object pairs as nodes and employing the pairwise spatial relations as node encoding. VSGNet et al. [41] leverages graph convolution and spatial configuration to refine visual features of human-object pairs and exploits structural connections between them. SCG [48] develops a bipartite graph to model interrelationships between nodes in HOI scene where each human node is connected to each object node. Building upon SCG, Park et al. [33] design a graph with a pose-conditioned self-loop structure to update the encoding of human nodes with local features of skeleton joints. Additionally, Zhang et al. [50] construct an interaction-centric graph by treating selected interaction proposals as graph nodes to examine inter-interaction semantic structure and intra-interaction spatial structure.

Recent advancements in HOI recognition tasks are also inspired by graphical models. LIGHTEN [40] employs a graph structure to model human and object embeddings, which serves them as nodes in the scene. In a similar vein, Dabral et al. [6] investigate the efficacy of GCNs in spatial relation learning compared to Convolutional Networks and Capsule Networks. Wang et al. [43] propose the STIGPN to understand the evolution of spatio-temporal relationships and distinguish the objects involved in the background using parsed graphs. Xing et al. [44] introduce a novel spatial attention mechanism that can enhance action recognition by adaptively generating a spatial-relation graph during HOIs. InterDiff [45] utilizes a diffusion model [4] combined with a physics-informed predictor to anticipate 3D HOIs, effectively capturing complex, long-term interactions by modeling dynamic objects and whole-body motion in a spatial-temporal graph neural network. In 2G-GCN [35], linking collective geometric features with individual visual features causes hierarchical misalignment, as high-level spatial information may not align well with detailed, entity-specific visual data. This focuses on less relevant objects and fails to explicitly learn HOIs. In this study, we develop an understanding of HOIs by progressing from category-level feature fusion to scenery-level graph representation, enabling a structured and thorough comprehension of interactions.

### 3 Methodology

We propose an end-to-end framework CATS (Fig. 1) to learn HOIs from category-level to scenery-level, which first focuses on the inherent characteristics of different categories, capturing their physical properties and contextual visual cues to achieve a rich feature representation. It then adopts a graph attention neural network to learn multi-category features as a scenery graph representation, which represents the true HOI. This approach mirrors natural cognitive processes [3, 26] facilitating a structured and comprehensive understanding of interactions within various contexts.

Alternative architecture performs suboptimally, an approach treats each human and object as an entity independently, ignoring the correlation between the same category and compromising the model’s ability to understand complex dynamics. An alternative method [35] groups all human poses and object bounding boxes into a single category for geometric feature learning, and then combines these geometric features with visual features in a single graph learning, which complicates entity representation and hampers explicit HOI learning. We compare these alternative architectures with our method in Experimental Results 4.



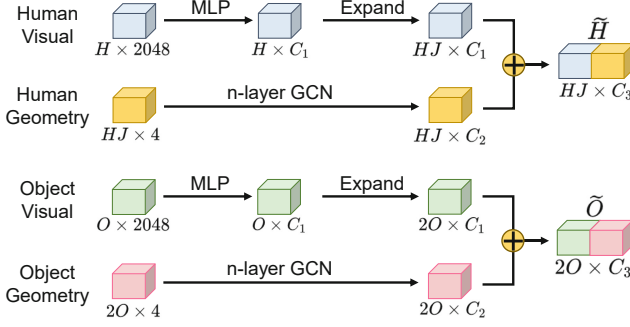
**Fig. 1.** Overview of our end-to-end framework CATS. We first learn geometric features via a graph for human and object categories, fusing them with corresponding visual features. Subsequently, a scenery interactive graph is constructed to deeply understand the interaction dynamics between multi-categories.

### 3.1 Multi-Category Multi-Modality Fusion

Previous CNN-based methods for HOI recognition in videos have predominantly focused on visual features [21, 29, 30], which may not be sufficient in cases of occlusion. While more advanced approaches like 2G-GCN [35] have attempted to incorporate geometric features to complement visual features, they categorize all human skeletons and object bounding boxes under a single category for geometric feature learning, thereby neglecting the distinct characteristics unique to each category and potentially generating skewed geometric features.

To this end, we propose a multi-category multi-modality fusion module that first learns geometric features via a graph for human and object two categories and then fuses them with corresponding visual features (Fig. 1). These category-specific features establish a rich multimodal context, providing a solid foundation for subsequent accurate interaction recognition.

**Geometric Features** For feature representation in human category and other related tasks, following previous successes [24, 35], we concatenate the position



**Fig. 2.** The process of learning and fusing geometric and visual features for human and object categories.

and velocity of all humans into keypoint channels, forming human geometric features  $\mathcal{HG} = \{hg_{t,h,j}\}_{t=1,h=1,j=1}^{T,H,J} \in \mathbb{R}^4$ , where  $hg_{t,h,j}$  denotes the body joint of type  $j$  in human  $h$  at time  $t$ ,  $T$  denotes the total number of frames in the video,  $H$  and  $J$  denote the total number of humans and keypoints of a human body in a frame, respectively. Similar to humans, object geometric features  $\mathcal{OG} = \{og_{t,o,u}\}_{t=1,o=1,u=1}^{T,O,2} \in \mathbb{R}^4$ , where  $og_{t,o,u}$  denotes the bounding box diagonal points  $u$  in object  $o$  at time  $t$  and  $O$  denotes the total number of objects.

As shown in Fig. 2, human and object geometric features are adopted n-layer GCNs to capture spatial dynamics and interactions in each category. This enables deeper analysis through successive transformations, allowing the graph-based network to learn intricate patterns of spatial dynamic interactions at multiple levels of abstraction [9, 46]. Here, taking human geometric features as an example, the operation of each GCN layer is formalized as:

$$H^{(l+1)} = \sigma \left( AH^{(l)}W^{(l)} \right), \quad (1)$$

where  $H^{(l)}$  represents the activation matrix at the  $l$ th layer ( $H^{(0)} = \mathcal{HG}$  for the initial layer),  $A$  is the adjacency matrix defining the graph structure,  $W^{(l)}$  is the weight matrix for the  $l$ th layer, and  $\sigma$  is the Tanh activation function.

For an n-layer GCN, this transformation is applied iteratively to obtain the final embedded human geometric features:

$$HG' = H^{(n)} = \sigma \left( AH^{(n-1)}W^{(n-1)} \right) \quad (2)$$

where  $n$  is the total number of GCN layers, iterating the process from  $l = 0$  to  $n - 1$ . We choose  $n = 4$  based on empirical experimental results. Through this operation, we can obtain the embedded human and object geometric features:  $HG' \in \mathbb{R}^{T \times HJ \times C_2}$  and  $OG' \in \mathbb{R}^{T \times 2O \times C_2}$ .

**Visual Features** In contrast to geometric features, visual features in videos offer a wealth of contextual information and essential feature representations.



Following [30,35], we derive 2048-dimensional visual features of entities from Region of Interest (ROI) pooled 2D bounding boxes around humans and objects in video frames. As shown in Fig. 2, they are subsequently reduced dimensionally to  $C_1$  through an MLP with learnable embeddings and aligned dimensionally with geometric features. This process results in the embedded human and object visual features:  $HV' \in \mathbb{R}^{T \times HJ \times C_1}$  and  $OV' \in \mathbb{R}^{T \times 2O \times C_1}$ .

**Multi-Modality Fusion** Finally, we fuse embedded geometric and visual features in the human and object keypoint channel, producing new enriched human and object feature representations, respectively:

$$\tilde{H} = HG' \oplus HV' \in \mathbb{R}^{T \times HJ \times C_3}, \quad (3)$$

$$\tilde{O} = OG' \oplus OV' \in \mathbb{R}^{T \times 2O \times C_3}, \quad (4)$$

where  $\oplus$  represents concatenate operation and  $C_3 = C_1 + C_2$ . This refined fusion of geometric and visual cues creates a richly contextualized blend, laying a solid foundation for enhanced scenery graph learning of HOIs.

### 3.2 Scenery Interactive Graph

To effectively model the interactions between humans and objects, the existing method [30] focuses exclusively on their visual features to construct an interaction graph. This approach taps into the visual aspect of interactions, which is essential but insufficient for grasping the dynamic spatial relationships critical to understanding the complexities of HOI. Furthermore, 2G-GCN [35] offers a more comprehensive view but fuse geometric features representing all entities with visual features representing individuals, which results in hierarchical misalignment and fails to explicitly learn HOIs.

To overcome the constraints of prior approaches, we propose a scenery interactive graph that adopts a graph attention neural network to learn interactions between different categories with enriched feature representation (Fig. 1), to deeply understand the interaction dynamics among all humans and objects. This structured approach facilitates a comprehensive understanding of interactions within various contexts.

**GAT for Learning Scenery Graph** Specifically, we adopt Graph Attention Networks (GAT) [14] in learning scenery graph interactions is particularly advantageous due to their ability to dynamically adjust to rapid changes in human and object interactions within scenery graphs, thanks to their adaptive edge weighting and handling of non-static features. This ensures a precise focus on relevant entities and their evolving relationships, optimizing the model’s responsiveness to the complex dynamics of interactions.

We construct the HOI scenery graph  $\mathcal{G}_{s-t} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} \in \mathbb{R}^{T \times (HJ+2O) \times C_3}$  represents the node features, which is obtained by concatenating the local human feature representation  $\tilde{H}$  and object feature representation  $\tilde{O}$ ,

and  $\mathcal{E} \in \mathbb{R}^{T \times (HJ+2O) \times (HJ+2O)}$  denotes the initialized fully-connected adjacency matrix. For each node  $\mathcal{V}_i$  at time step  $t \in [1, \dots, T]$ , the feature representation is:

$$\mathcal{V}_i^t = \sigma \left( \sum_{j \in \mathcal{N}(i) \cup i} \alpha_{i,j}^t \Theta \mathcal{V}_j^t \right), \quad (5)$$

and the attention coefficients  $\alpha_{i,j}$  are computed as:

$$\alpha_{i,j}^t = \frac{\exp(\text{LeakyReLU}(\mathbf{W}^\top [\Theta \mathcal{V}_i^t, \|\cdot\|, \Theta \mathcal{V}_j^t]))}{\sum_{n \in \mathcal{N}(i) \cup i} \exp(\text{LeakyReLU}(\mathbf{W}^\top [\Theta \mathcal{V}_i^t, \|\cdot\|, \Theta \mathcal{V}_n^t]))}, \quad (6)$$

where  $\Theta(\cdot)$  is the transformation function,  $\mathcal{N}(\cdot)$  is the neighbor set of node  $i$  and  $\mathbf{W}$  represents learnable parameters. This dynamic weighting is crucial as it allows the model to adaptively focus on the most relevant nodes and edges, reflecting the changing nature of interactions and relationships within the scene.

**RNN-based Network for Learning Temporal Dependency** After obtaining the learned HOI scenery graph representations at each time step  $t$ , we employ an RNN-based network to learn the temporal dependencies across all the time steps. Specifically, we utilize a Bi-direction Gated Recurrent Unit (Bi-GRU) [5] that enables our model to integrate both past and future contexts, enhancing its understanding of the sequential dynamics in human-object interactions. The GRU’s gating mechanisms effectively manage long-term dependencies, ensuring robust temporal modeling. For the learned step-wise feature representations, we utilize a Gumbel-Softmax module [16], enabling precise and adaptable delimitation of sub-event lengths in video sequences. This module is instrumental in enabling gradient-based optimization while maintaining probabilistic integrity in segmenting actions, a crucial aspect when dealing with the inherently fluctuating characteristics of video content. Subsequently, we employ another Bi-GRU to discern the temporal relations among segmented sub-actions. The processed features are then leveraged to identify specific sub-activities associated with humans, with the granularity of recognition tailored to suit the requirements of the specific dataset.

## 4 Experiments

### 4.1 Datasets

We evaluate CATS on two datasets: MPHUI-72 [35] and CAD-120 [18], showcasing the superior results on multi-person and single-person HOI recognition.

The MPHUI-72 dataset is valuable for two-person HOI tasks. It contains 72 videos of 8 pairs of people performing 3 distinct activities (*Cheering*, *Hair cutting* and *Co-working*) with 13 human sub-activities (e.g., *Sit*, *Pour*). Each video showcases two participants interacting with 2-4 objects from 3 unique angles. Geometric features and human sub-activities labels are frame-wise annotated.

CAD-120 is a prominent dataset for single-person HOI recognition. It contains 120 RGB-D videos, capturing 10 distinct activities executed by 4 participants, each repeated three times. In each video, a participant interacts with 1-5 objects. The dataset provides frame-wise annotations for 10 human sub-activities (e.g., *opening*, *placing*).

## 4.2 Evaluation Protocol

Following the evaluation protocol of [30,35], we assess CATS across two specific tasks: joint segmentation and label recognition for pre-segmented entities. The initial task involves both segmenting and classifying the timeline of each entity in a video, while the second extends this by assigning labels to pre-segmented sections with known ground truth. We adopt the F1@ $k$  metric [22] for evaluation, using standard thresholds of  $k = 10\%$ ,  $25\%$ , and  $50\%$ . This metric, prevalent in segmentation research [10,22,30], determines the correctness of a predicted action segment based on its minimum Intersection over Union (IoU) overlap with the ground truth and is particularly effective for assessing brief actions and detailed segmentation. For dataset evaluation, we implement a leave-two-subjects-out strategy for the MPHUI-72 dataset and a leave-one-subject-out cross-validation approach for CAD-120.

## 4.3 Network Setting

The visual features of humans and objects are extracted from 2D bounding boxes within the video using a Faster R-CNN module [36] that has been pre-trained [2] on the Visual Genome dataset [20]. For multi-modality fusion, we set  $C_1 = 512$  and  $C_2 = 256$ , resulting in a fused dimension of  $C_3 = 768$ , which supports varied feature dimensions as shown in Fig. 2.

## 4.4 Quantitative Comparison

**Multi-person HOIs** In the MPHUI-72 dataset, results in Table 1 demonstrate CATS not only surpasses the previous state-of-the-art models, ASSIGN [30] and 2G-GCN [35], showcasing significant performance improvements, but also exhibits unparalleled stability. This is highlighted by CATS’s superior performance across all F1 configurations coupled with substantially lower standard deviations. Specifically, in the F1@10 score, CATS achieves 71.3%, which is approximately 3% and 12% higher than 2G-GCN and ASSIGN, respectively, marking a clear advancement in both predictive accuracy and consistency in the domain of human-object interaction recognition. These experimental outcomes further underscore the significance of geometric features in the multi-person Human-Object Interaction (MPHUI) domain. Models based solely on visual features, such as ASSIGN, are noticeably outperformed by those that incorporate both visual and geometric information. Although 2G-GCN integrates both visual and geometric features, its sub-optimal performance can be attributed to a lack

of specificity in representing individual entities. Consequently, our model’s superior performance and stability are not just a result of integrating multiple types of features but also our model’s ability to specifically and effectively capture the nuanced dynamics of each entity involved in the interaction.

**Table 1.** Joined segmentation and label recognition on MPHUI-72.

Model	Sub-activity		
	F <sub>1</sub> @10	F <sub>1</sub> @25	F <sub>1</sub> @50
ASSIGN [30]	59.1 ± 12.1	51.0 ± 16.7	33.2 ± 14.0
2G-GCN [35]	68.6 ± 10.4	60.8 ± 10.3	45.2 ± 6.5
CATS	<b>71.3 ± 5.0</b>	<b>65.8 ± 3.9</b>	<b>48.8 ± 5.3</b>

**Single-person HOIs** In the CAD-120 dataset, as presented in Table 2, CATS demonstrates strong competitiveness in the single-person HOI scenarios. For both human sub-activity and object affordance labelling tasks, CATS surpasses various prior methods, including those reliant on visual features like ATCRF [17] and [30], as well as the more sophisticated visual-geometric approach offered by 2G-GCN [35]. Notably, CATS secures SOTA performance in both F1@10 and F1@25 metrics, registering improvements of 1.6% and 0.1% over ASSIGN and 2G-GCN, respectively. This achievement underscores CATS’s exceptional capability to accurately model and predict the dynamics of interactions, highlighting its adaptability and efficiency across different HOI challenges.

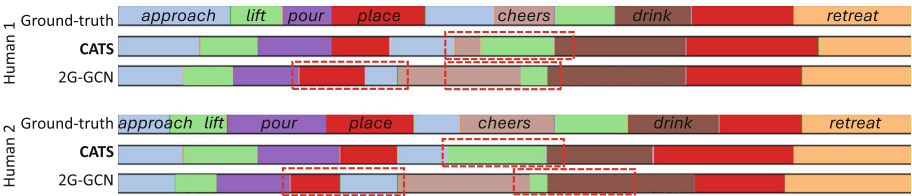
**Table 2.** Joined segmentation and label recognition on CAD-120.

Model	Sub-activity		
	F <sub>1</sub> @10	F <sub>1</sub> @25	F <sub>1</sub> @50
rCRF [38]	65.6 ± 3.2	61.5 ± 4.1	47.1 ± 4.3
Independent BiRNN	70.2 ± 5.5	64.1 ± 5.3	48.9 ± 6.8
ATCRF [17]	72.0 ± 2.8	68.9 ± 3.6	53.5 ± 4.3
Relational BiRNN	79.2 ± 2.5	75.2 ± 3.5	62.5 ± 5.5
ASSIGN [30]	88.0 ± 1.8	84.8 ± 3.0	73.8 ± 5.8
2G-GCN [35]	89.5 ± 1.6	87.1 ± 1.8	<b>76.2 ± 2.8</b>
CATS	<b>89.6 ± 2.1</b>	<b>87.3 ± 1.5</b>	76.0 ± 3.5

#### 4.5 Qualitative Comparison

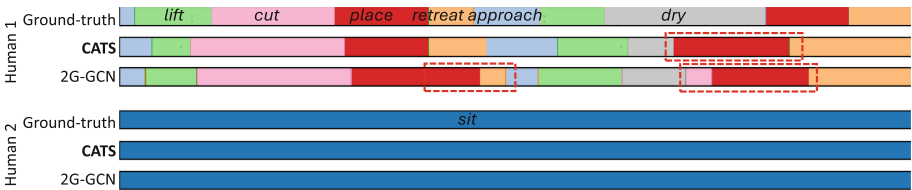
In this section, we present a qualitative comparison of CATS with the state-of-the-art method across the MPHUI-72 and CAD-120 datasets.

Fig. 3 and Fig. 4 illustrate *Cheering* and *Hair Cutting* activities within the MPHOI-72 dataset, comparing the segmentation and labeling tasks performed by CATS and 2G-GCN [35] against the ground truth. Significant segmentation errors are marked with red dashed boxes. Although both methods exhibit some discrepancies in their predictions, CATS more closely aligns with the ground truth, offering a more precise and stable visualization across a variety of actions. Conversely, 2G-GCN is prone to generating inappropriate sub-activities such as *cheers* and *lift* in the *Cheering* activity. Moreover, in the *Hair Cutting* activity, 2G-GCN inaccurately presents the *cut* sub-activity into *place* sub-activity, further deviating from the expected interaction dynamics. This comparison underscores the superior accuracy and reliability of CATS in capturing and visualizing complex human-object interactions within diverse scenarios.



**Fig. 3.** Visualization of segmentation on MPHOI-72 for *Cheering* activity. Red dashed boxes highlight major segmentation errors. (Color figure online)

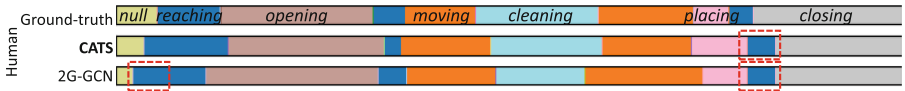
Fig. 5 and Fig. 6 illustrate the *Cleaning Objects* and *Making Cereal* activities from the single-person CAD-120 dataset, with abnormal segmentation instances accentuated by red dashed boxes. For the *Cleaning Objects* activity, both methods effectively match the overall ground truth. However, CATS provides a visualization that more closely approximates the ground truth. In the *Making Cereal* activity, CATS significantly outperforms 2G-GCN, particularly in sub-activities such as *pouring*, *moving*, and *reaching*, while 2G-GCN yields some inaccurate segmentations. The enhanced precision of CATS in capturing the intricacies of each activity highlights its superior performance, excelling in the identification and precise representation of detailed actions and interactions within the scenes, thus delivering a more accurate and reliable analysis of the activities performed.



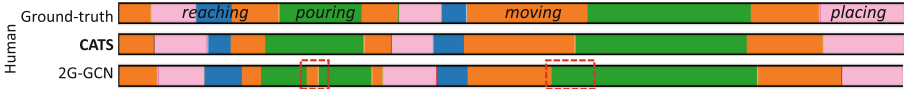
**Fig. 4.** Visualization of segmentation on MPHOI-72 for *Hair cutting* activity. Red dashed boxes highlight major segmentation errors.

## 4.6 Alternative Architectures and Ablation Studies

**Architecture Alternatives Comparison** We evaluate the HOI recognition performance on the MPHUI-72 and CAD-120 datasets by conducting tests on various alternative model structures. The experimental outcomes, as detailed in Tables 3 and 4, reveal that our model consistently delivers superior results compared to these alternatives. This superior performance is likely attributable to the unique consideration our model gives to category-level interactions, specifically the distinct analysis of human-human and object-object interactions. Unlike other approaches that might treat interactions generically or overlook the nuanced distinctions between different types of interactions, our model maintains a comprehensive view.



**Fig. 5.** Visualization of segmentation on CAD-120 for *Cleaning objects* activity. Red dashed boxes highlight major segmentation errors. (Color figure online)



**Fig. 6.** Visualization of segmentation on CAD-120 for *Making Cereal* activity. Red dashed boxes highlight major segmentation errors. (Color figure online)

**Table 3.** Comparison between architecture alternatives and CATS on MPHUI-72.

Model	Sub-activity		
	F <sub>1</sub> @10	F <sub>1</sub> @25	F <sub>1</sub> @50
Independent-entity architecture	65.1 ± 3.3	58.7 ± 1.7	40.4 ± 3.9
2G-GCN [35]	68.6 ± 10.4	60.8 ± 10.3	45.2 ± 6.5
CATS	71.3 ± 5.0	65.8 ± 3.9	48.8 ± 5.3

**Table 4.** Comparison between architecture alternatives and CATS on CAD-120.

Model	Sub-activity		
	F <sub>1</sub> @10	F <sub>1</sub> @25	F <sub>1</sub> @50
Independent-entity architecture	85.9 ± 4.0	84.1 ± 4.9	72.8 ± 5.2
2G-GCN [35]	89.5 ± 1.6	87.1 ± 1.8	76.2 ± 2.8
CATS	89.6 ± 2.1	87.3 ± 1.5	76.0 ± 3.5

**GCN Layers for Geometric Feature Learning** In this section, we conduct ablation studies to elucidate the impact of the depth of GCN layers on the geometric learning of human joints and object keypoints within our network, results are shown in Table 5. To assess the influence of GCN layer depth on model performance, we explore configurations with 1, 2, 3, 4, and 5 GCN layers. Through this comparative analysis, we aim to identify the most effective layer

**Table 5.** Results of different GCN layers in multi-category multi-modality fusion on MPHOI-72.

Model	Sub-activity		
	F <sub>1</sub> @10	F <sub>1</sub> @25	F <sub>1</sub> @50
1-layer GCN	70.4 ± 1.7	62.0 ± 2.5	43.9 ± 3.8
2-layer GCN	68.8 ± 4.3	62.1 ± 4.3	44.0 ± 3.3
3-layer GCN	67.4 ± 4.2	63.3 ± 3.4	44.2 ± 1.3
5-layer GCN	70.4 ± 5.7	60.0 ± 2.3	43.7 ± 2.2
4-layer GCN (Ours)	<b>71.3 ± 5.0</b>	<b>65.8 ± 3.9</b>	<b>48.8 ± 5.3</b>

depth that balances computational efficiency with the nuanced understanding of spatial relationships essential for interpreting complex interactions between humans and objects. The results indicate that a configuration of 4-layer GCN offers the optimal balance, providing the best performance in terms of both accuracy and computational efficiency. This depth allows for sufficient complexity to understand and model the geometric relationships critical for accurate interaction recognition, without incurring the diminishing returns or increased computational demand associated with additional layers.

## 5 Conclusion

In conclusion, we propose CATS, an advanced end-to-end framework that enhances video-based HOI recognition through sophisticated integration of category and scenery level analyses. It first fuses multi-modal features of different categories, and then construct a scenery interactive graph to learn the relationships between these categories. CATS demonstrates superior performance on key benchmarks such as MPHOI-72 and CAD-120 datasets, showcasing the effectiveness of multi-person and single-person HOI recognition.

**Acknowledgement.** This research is supported in part by the EPSRC NorthHFutures project (ref: EP/X031012/1).

## References

1. Agarwal, A., Dabral, R., Jain, A., Ramakrishnan, G.: Skew-robust human-object interactions in videos. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5098–5107 (2023)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR. pp. 6077–6086 (2018)
3. Baldassano, C., Beck, D.M., Fei-Fei, L.: Human-object interactions are more than the sum of their parts. *Cereb. Cortex* **27**(3), 2276–2288 (2017)

4. Chang, Z., Koulieris, G.A., Shum, H.P.H.: On the design fundamentals of diffusion models: A survey. arXiv (2023). <https://doi.org/10.48550/arXiv.2306.04542>
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint [arXiv:1412.3555](https://arxiv.org/abs/1412.3555) (2014)
6. Dabral, R., Sarkar, S., Reddy, S.P., Ramakrishnan, G.: Exploration of spatial and temporal modeling alternatives for hoi. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2281–2290 (2021)
7. Dogariu, M., Stefan, L.D., Constantin, M.G., Ionescu, B.: Human-object interaction: Application to abandoned luggage detection in video surveillance scenarios. In: 2020 13th International Conference on Communications (COMM). pp. 157–160. IEEE (2020)
8. Dreher, C.R., Wächter, M., Asfour, T.: Learning object-action relations from bimanual human demonstration using graph networks. IEEE Robotics and Automation Letters **5**(1), 187–194 (2020)
9. Du, S.S., Hou, K., Salakhutdinov, R.R., Poczos, B., Wang, R., Xu, K.: Graph neural tangent kernel: Fusing graph neural networks with graph kernels. NeurIPS **32** (2019)
10. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: CVPR. pp. 3575–3584 (2019)
11. Gao, C., Xu, J., Zou, Y., Huang, J.B.: Drg: Dual relation graph for human-object interaction detection. In: ECCV. pp. 696–712 (2020)
12. Gao, C., Zou, Y., Huang, J.B.: ican: Instance-centric attention network for human-object interaction detection. arXiv preprint [arXiv:1808.10437](https://arxiv.org/abs/1808.10437) (2018)
13. Gkioxari, G., Girshick, R., Malik, J.: Actions and attributes from wholes and parts. In: ICCV. pp. 2470–2478 (2015)
14. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In: ICCV. pp. 6272–6281 (2019)
15. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR. pp. 5308–5317 (2016)
16. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint [arXiv:1611.01144](https://arxiv.org/abs/1611.01144) (2016)
17. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE TPAMI **38**(1), 14–29 (2016)
18. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. The International Journal of Robotics Research **32**(8), 951–970 (2013)
19. Krebs, F., Meixner, A., Patzer, I., Asfour, T.: The kit bimanual manipulation dataset. In: IEEE/RAS International Conference on Humanoid Robots (Humanoids). pp. 0–0 (2021)
20. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV **123**(1), 32–73 (2017)
21. Le, H., Sahoo, D., Chen, N.F., Hoi, S.C.: Bist: Bi-directional spatio-temporal reasoning for video-grounded dialogues. arXiv preprint [arXiv:2010.10095](https://arxiv.org/abs/2010.10095) (2020)
22. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR. pp. 156–165 (2017)
23. Li, L., Shum, H.P.H., Breckon, T.P.: Less is More: Reducing Task and Model Complexity for 3D Point Cloud Semantic Segmentation. In: CVPR (2023)
24. Li, L., Shum, H.P.H., Breckon, T.P.: RAPiD-Seg: Range-Aware Pointwise Distance Distribution Networks for 3D LiDAR Segmentation. In: ECCV (2024)







25. Li, R., Katsigiannis, S., Shum, H.P.: Multiclass-sgcn: Sparse graph-based trajectory prediction with agent class embedding. In: ICIP. pp. 2346–2350. IEEE (2022)
26. Li, Y.L., Liu, X., Wu, X., Li, Y., Lu, C.: Hoi analysis: Integrating and decomposing human-object interaction. *NeurIPS* **33**, 5011–5022 (2020)
27. Liu, M., Tang, S., Li, Y., Rehg, J.M.: Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In: ECCV. pp. 704–721. Springer (2020)
28. Mallya, A., Lazebnik, S.: Learning models for actions and person-object interactions with transfer to question answering. In: ECCV. pp. 414–428 (2016)
29. Maraghi, V.O., Faez, K.: Zero-shot learning on human-object interaction recognition in video. In: 2019 5th Iranian conference on signal processing and intelligent systems (ICSPIS). pp. 1–7 (2019)
30. Morais, R., Le, V., Venkatesh, S., Tran, T.: Learning asynchronous and sparse human-object interaction in videos. In: CVPR. pp. 16041–16050 (2021)
31. Mukherjee, D., Gupta, K., Chang, L.H., Najjaran, H.: A survey of robot learning strategies for human-robot collaboration in industrial settings. *Robotics and Computer-Integrated Manufacturing* **73**, 102231 (2022)
32. Nagarajan, T., Feichtenhofer, C., Grauman, K.: Grounded human-object interaction hotspots from video. In: ICCV. pp. 8688–8697 (2019)
33. Park, J., Park, J.W., Lee, J.S.: Viplo: Vision transformer based pose-conditioned self-loop graph for human-object interaction detection. In: CVPR. pp. 17152–17162 (2023)
34. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: ECCV. pp. 401–417 (2018)
35. Qiao, T., Men, Q., Li, F.W.B., Kubotani, Y., Morishima, S., Shum, H.P.H.: Geometric features informed multi-person human-object interaction recognition in videos. In: ECCV (2022)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE TPAMI* **39**(6), 1137–1149 (2016)
37. Rezaee, K., Rezakhani, S.M., Khosravi, M.R., Moghimi, M.K.: A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. *Pers. Ubiquit. Comput.* **28**(1), 135–151 (2024)
38. Sener, O., Saxena, A.: rcrf: Recursive belief estimation over crfs in rgb-d activity videos. In: *Robotics: Science and systems*. Citeseer (2015)
39. Smith, B.A., Yin, Q., Feiner, S.K., Nayar, S.K.: Gaze locking: passive eye contact detection for human-object interaction. In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. pp. 271–280 (2013)
40. Sunkesula, S.P.R., Dabral, R., Ramakrishnan, G.: Lighten: Learning interactions with graph and hierarchical temporal networks for hoi in videos. In: *ACM MM*. pp. 691–699 (2020)
41. Ulutan, O., Iftekhar, A.S.M., Manjunath, B.S.: Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. In: CVPR. pp. 13617–13626 (2020)
42. Wang, H., Zheng, W.s., Yingbiao, L.: Contextual heterogeneous graph network for human-object interaction detection. In: ECCV. pp. 248–264 (2020)
43. Wang, N., Zhu, G., Zhang, L., Shen, P., Li, H., Hua, C.: Spatio-temporal interaction graph parsing networks for human-object interaction recognition. In: *ACM MM*. pp. 4985–4993 (2021)
44. Xing, H., Burschka, D.: Understanding spatio-temporal relations in human-object interaction using pyramid graph convolutional network. In: *2022 IEEE/RSJ Inter-*

- national Conference on Intelligent Robots and Systems (IROS). pp. 5195–5201 (2022)
45. Xu, S., Li, Z., Wang, Y.X., Gui, L.Y.: Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14928–14940 (2023)
  46. You, Y., Chen, T., Wang, Z., Shen, Y.: L2-gcn: Layer-wise and learned efficient training of graph convolutional networks. In: CVPR. pp. 2127–2135 (2020)
  47. Zeng, Z., Dai, P., Zhang, X., Zhang, L., Cao, X.: Cognition guided human-object relationship detection. *IEEE Transactions on Image Processing* (2023)
  48. Zhang, F.Z., Campbell, D., Gould, S.: Spatially conditioned graphs for detecting human-object interactions. In: ICCV. pp. 13319–13327 (2021)
  49. Zhang, M., Wu, X., Yuan, Z., He, Q., Huang, X.: Human-object-object interaction: Towards human-centric complex interaction detection. In: ACM MM. pp. 2233–2242 (2023)
  50. Zhang, Y., Pan, Y., Yao, T., Huang, R., Mei, T., Chen, C.W.: Exploring structure-aware transformer over interaction proposals for human-object interaction detection. In: CVPR. pp. 19548–19557 (2022)
  51. Zhuo, T., Cheng, Z., Zhang, P., Wong, Y., Kankanhalli, M.: Explainable video action reasoning via prior knowledge and state transitions. In: ACM MM. pp. 521–529 (2019)



# Adaptive Global Gesture Paths and Signature Features for Skeleton-based Gesture Recognition

Dongzi Shi<sup>1</sup> , Xin Zhang<sup>1</sup> , Jiale Cheng<sup>1,2</sup> , Tong Xiong<sup>1</sup>, and Hao Ni<sup>3</sup> 

<sup>1</sup> South China University of Technology, Guangzhou, Guangdong 510000, China  
{eexinzhang, eedongzishi}@mail.scut.edu.cn

<sup>2</sup> University of North Carolina at Chapel Hill, Chapel Hill, NC 27514, USA

<sup>3</sup> University College London, London WC1E6BT, UK  
h.ni@ucl.ac.uk

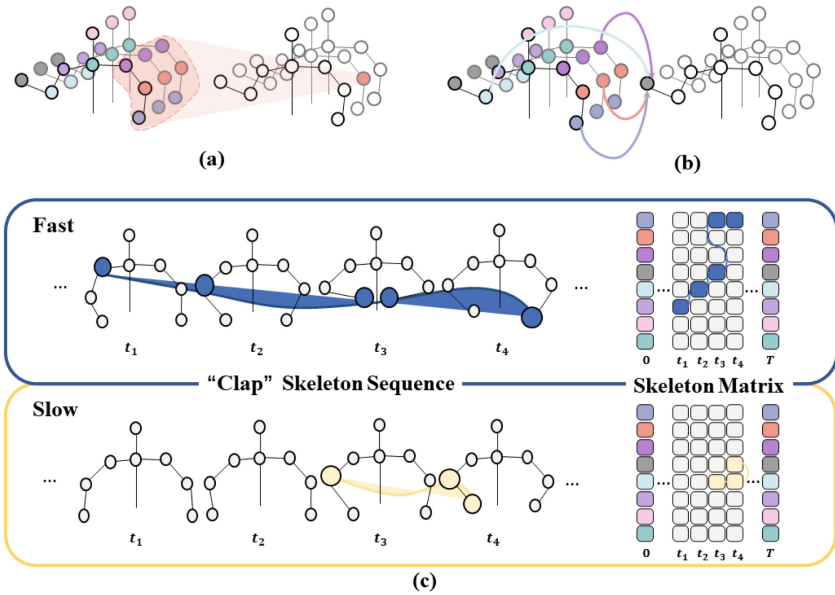
**Abstract.** Gestures exhibit sparse joint variations and different time scales, making local dynamic analysis and global spatio-temporal modeling important. Path signature provides mathematical and dynamic analysis of joint trajectories to assist in spatio-temporal modeling. However, previous methods relied on predefined local spatio-temporal joint trajectories, also known as paths. This limitation makes it challenging to directly capture the dynamics of the entire gesture and adapt to varying scales of gesture changes. In this work, we construct the Adaptive Global Gesture Path and extract its signature features as gesture representations. Specifically, we designed global branch to model the global spatio-temporal variation relationship of joints. The dynamic branch is based on the proposed Motion Guided Cluster Attention Block, which emphasizes joints exhibiting similar motion patterns. Combining two branches, the predicted dynamic and global score can distinguish key joints at different times to construct the Adaptive Global Gesture Path that condensely represents the entire gesture. We conducted experiments on the ChaLearn2013 and WLASL datasets, and achieved the state-of-the-art results with much smaller model size.

**Keywords:** Skeleton-based gesture recognition · Attention mechanism · Path signature

## 1 Introduction

Gesture recognition is an active research area for its wide range of applications in human-computer interaction [36] and sign language translation [1]. The advancement of pose estimation methods [2, 14] has facilitated the acquisition of skeleton-based gesture data, which can be represented as a spatio-temporal matrix as shown in Figure 1 (c). In contrast to RGB-based approaches, skeleton-based methods are more robust to noise, occlusion, and viewpoint changes.

The key to skeleton-based gesture recognitions lies in the learning of gesture representation that captures spatio-temporal interactions and temporal

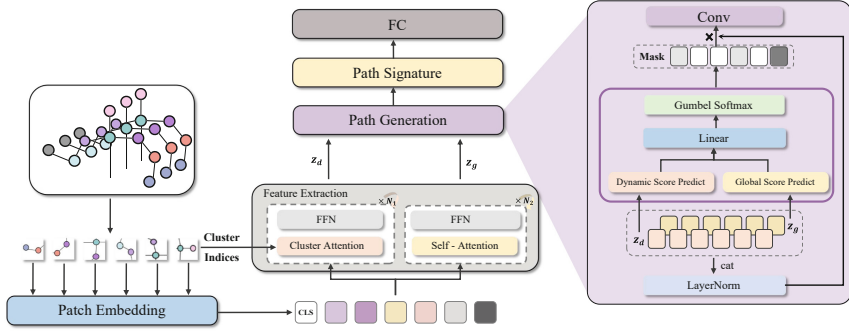


**Fig. 1.** Illustration of joints interaction and connection. (a) (b) shows the differences between regular convolution and the proposed Motion-Guided Cluster Attention in associating joints. (c) compares skeleton sequences execution speeds of the “clap” gesture and the corresponding paths.

dynamic analysis. Existing CNNs [10,30] and GCNs methods [4,34,48] exhibit constraints in effectively capturing interaction among distant key joints. For instance, the distance between the left and right hand in the spatio-temporal matrix exceeds the receptive field of a single convolutional operation. Consequently, the association between these joints is established only in deeper layers. Some work [37,38] captures spatio-temporal features by deep Transformer structure with global receptive fields, but important local dynamic descriptions cannot be emphasized.

Recently, path signature based action recognition [24,30,31] have demonstrated exceptional performance due to its temporal dynamic analysis capabilities and effective motion dependency description. The sequential arrangement of skeleton data inherently embodies a natural path-like structure. By constructing effective paths and integrating path signature, these approaches successfully capture the geometric and analytical attributes inherent in gesture trajectories. However, existing methods concentrate on predefined connection patterns, making the dynamic analysis between unconnected joints and the overall evolution and patterns of gestures at different time scales challenging.

In this work, we define a spatio-temporal joint trajectory referred as Adaptive Global Gesture Path, which condensely represents the entire gesture. The key challenge lies in extracting global spatio-temporal features and dynamic



**Fig. 2.** The overall framework of the proposed approach. The two branches extract features separately, subsequently integrating the dynamic and global features to identify key tokens with predicted score. Finally, we obtain the spatio-temporal path of the gesture and extract signature features for recognition.

interaction to distinguish key joints in different gestures with sparse variations. The entire process can be divided into three steps (Fig. 2).

First, we model spatio-temporal dependencies and temporal dynamics through global and dynamic branches. The proposed Motion-Guided Cluster Attention Block breaks away from fixed patterns, allowing the model to adeptly capture temporal dynamic interactions on a global scale.

Next, we evaluate the significance based on the information within the joints, sampling key joints to construct the Adaptive Global Gesture Path. Inspired by DynamicViT [39], we predict a dynamic and a global score to quantify the significance of joints. Gumbel-Softmax [20] is then employed to overcome the non-differentiable problem in sampling with a generated binary mask.

Lastly, we incorporate the rough path theory to extract the signature features of the Adaptive Global Gesture Path. Path signatures exhibit numerous algebraic and analytical properties, providing a comprehensive representation that effectively captures complex interactions over time.

In general, we conclude our contributions in three aspects:

- We expand the definition of gesture paths and construct a simple and condensed spatial-temporal trajectory to represent the gesture. We refer that trajectory as Adaptive Global Gesture Path, and further, its signature features are obtained for the final recognition process.
- We propose a Motion Guided Cluster Attention Block for dynamic analysis, which breaks down predefined forms and definitions, and aggregates joint information with similar motion.
- We perform extensive experiments on ChaLearn2013 and WLASL-300. Our proposed method has significantly improved accuracy, with smaller model sizes. Ablation study further validated the usefulness of the proposed block and path signature features.

## 2 Related Work

### 2.1 Skeleton-based Gesture Recognition

Recent work have focused on extracting more discriminative spatio-temporal features. On one hand, previous work [28,32] successfully incorporated global information to facilitate network adaptability at various time scales. [22] emphasized the association of distant nodes and proposed a hierarchical decomposition graph. On the other hand, it is evident that not all joint information contributes equally to recognition task. Therefore, [42,46] developed spatial-temporal attention mechanisms to identify the most informative joints at different temporal instances. These advancements have significantly enhanced the discriminative power and efficiency of gesture recognition models. Furthermore, the incorporation of motion information, extracted from the skeleton coordinate sequences, has been widely adopted in skeleton-based action recognition studies [11,27] and skeleton-based gesture recognition research [5,32]. [23] emphasize the modeling of partial motion context information and integrate each part into a unified context. Building upon these advancements, our work also leverages motion information to guide the network in adapting its joint interactions dynamically.

### 2.2 Path Signature Method

Path signature was first proposed by Chen [3] as an infinite and graded sequence of iterated integrals of a path of bounded variation. The signature can characterise the path up to a negligible re-parameterization equivalence. Lyons [17] extended it on this basis to apply it to finite p-variational rough paths. Afterwards, path signature combined with machine learning was successfully applied in various fields, including financial data analysis [35], handwritten character recognition [47], author recognition [21], infant cognitive score prediction [6,52], and skeleton based action recognition [24,30,31,49].

Previous path signature based action recognition methods were limited to associating adjacent joints locally in both time and space, failing to capture the entirety of the action’s execution process. In this work, the network adaptively selects key joints on a global scale, directly constructing a more condensed and representative spatial-temporal joined path spanning the entire motion process.

### 2.3 Vision Transformer

Motivated by the powerful contextual modeling ability of the Transformer [44], there is an emerging research area to employ the Transformer for various applications in the field of computer vision [8,18,33]. However, the Transformer introduces a significant amount of computation, and in visual tasks, which usually contains a lot of redundant information. Therefore, many works focused on sparsifying ViT [13,29,39,50] and removing redundant tokens, while [51] proposed to cluster and merge similar tokens to form a more flexible ViT. Additionally, as the model depth increases, the similarity between different blocks and tokens

also increases[15]. Hence, many works aimed to optimize this through multi-scale sequence interaction, improving attention mechanisms and forms, and increasing token diversity.

### 3 Methods

#### 3.1 Feature Extraction

The global branch captures the spatio-temporal features of gestures by establishing direct interaction between two joints. The dynamic branch takes into account joints with similar movement trajectories during gesture execution, thus better emphasizing local dynamics. By combining two branches, we obtain discriminative joint features.

The skeleton sequence can be viewed as a spatio-temporal matrix  $\mathbf{x} \in R^{C \times J \times T}$ , where  $C$  denotes the number of dimension,  $J$  is the number of joints,  $T$  is the number of frames. Given a patch size of  $(\Delta J, \Delta T)$ , we embed  $\mathbf{x}$  into a sequence of flattened 2D tokens  $\mathbf{z} \in R^{L \times D}$ , where  $L = \frac{J}{\Delta J} \times \frac{T}{\Delta T}$  represents the number of tokens, and  $D$  represents the embedded dimension of each token. A learnable embedding  $\mathbf{z}_{\text{cls}}$  termed class token is concatenated to the sequence.  $\mathbf{z} = [\mathbf{z}_{\text{cls}}, \mathbf{z}_1, \dots, \mathbf{z}_L]$  is then fed into two branches.

The global branch uses Self-Attention and FeedForward Neural Network (FFN) [44] to extract global feature  $\mathbf{z}_g$ :

$$\mathbf{z}_g = \text{Softmax}(\mathbf{Q}_g \mathbf{K}_g^T / \sqrt{D}) \mathbf{V}_g, \quad (1)$$

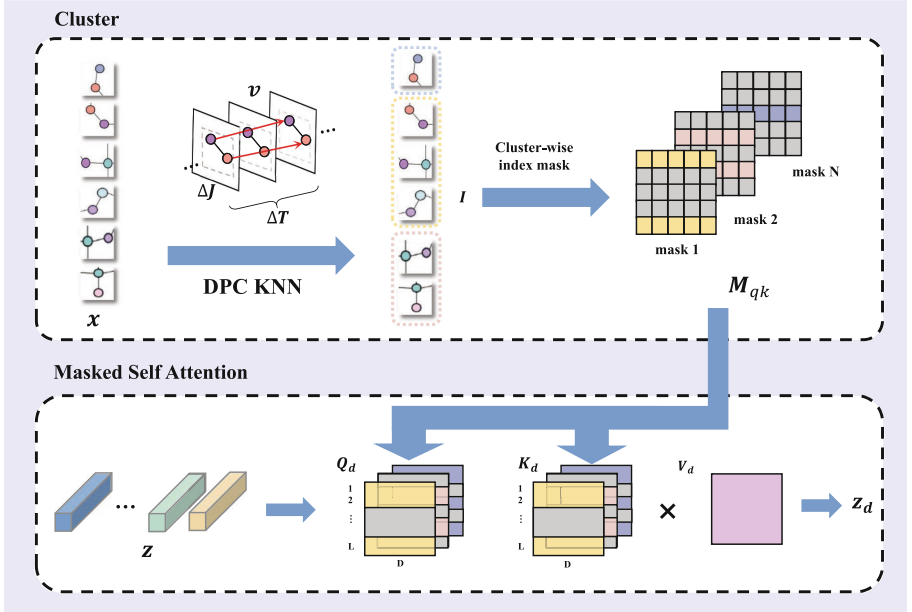
where  $\mathbf{Q}_g, \mathbf{K}_g, \mathbf{V}_g \in R^{L \times D}$  are obtained through linear mapping of the input  $\mathbf{z}$ .

The dynamic branch is composed of Motion-Guided Cluster Attention Block. As shown in Fig.3, the Motion-Guided Cluster Attention Block has two main process, token clustering and masked attention. Among them, we use the motion information of tokens to group them into  $N$  clusters, and generate mask based on the clustering results.

#### Motion-Guided Cluster Attention Block

Firstly, we calculate the velocity of the joints by measuring the difference in their coordinates over a time interval of  $\Delta T$  frames. We then use the DPC KNN algorithm[9] to cluster these velocities, resulting in a set of clustered indices  $I = \{i_1, \dots, i_L\}, i_l \in [1, N]$ .

Secondly, we mask tokens that do not belong to the same cluster and adaptively link relevant tokens. Specifically, we create a mask  $\mathbf{M}qk \in \{0, 1\}$  based on the cluster indices  $I$  for  $N$  distinct clusters. Then, we perform an element-wise multiplication between  $\mathbf{Q}d, \mathbf{K}d$  and the generated cluster mask  $\mathbf{M}qk$ , as shown in Eq. (2). This operation can be regarded as a flexible token-wise multi-head attention mechanism. Additionally, all attention mechanisms used in this paper are based on the multi-head attention mechanism.



**Fig. 3.** The process of clustering and masking out irrelevant tokens to calculate attention scores.

$$\begin{aligned}
 \hat{Q}_d &= M_{qk} * Q_d \\
 \hat{K}_d &= M_{qk} * K_d \\
 z_d &= \text{Softmax}(\text{Sum}(\hat{Q}_d \hat{K}_d^T / \sqrt{D})) V_d,
 \end{aligned} \tag{2}$$

where  $M_{qk} \in R^{N \times L \times D}$ ,  $Q_d, K_d, V_d$  are obtained through linear projection of the input  $z$ .

Motion-guided cluster attention breaks fixed information aggregation, targeting global-scale dynamic interactions. By adjusting  $N$ , we control the range of relevant tokens, and enable fine-grained attention.

### 3.2 Path Generation

By identifying key parts and connecting them, we construct a natural path-like structure that condenses the sequential skeletal gesture into a compact representation.

The probability of token retention is considered by both global and dynamic features. We calculate the similarity between class token  $z_{cls}$  and other tokens then followed by a linear layer to generate global prediction score  $s_g \in R^L$ :

$$s_g = (z_{cls} \mathbf{W}_q)(z_g \mathbf{W}_k)^T, \tag{3}$$



where  $\mathbf{W}_q, \mathbf{W}_k \in R^{D \times D}$  represents the learnable parameter used for linear projection.

Dynamic prediction score  $\mathbf{s}_d \in R^L$  is obtained by directly passing  $\mathbf{z}_d$  through an MLP module:

$$\mathbf{s}_d = \text{MLP}(\mathbf{z}_d). \quad (4)$$

Dynamic and global prediction scores are concatenated and fed into a linear projection to obtain the final prediction score  $\mathbf{s} \in [0, 1]^L$ :

$$\mathbf{s} = \text{Sigmoid}(\text{Linear}([\mathbf{s}_g, \mathbf{s}_d])). \quad (5)$$

Following that, we implemented a differentiable token sampling process using Gumbel-Softmax [20]. The temperature coefficient of the Gumbel-Softmax is gradually decay during training, so that the generated mask  $\mathbf{M} \in \{0, 1\}^L$  will gradually approach 0 and 1 during the training process, making the entire training process more stable:

$$\mathbf{M} = \text{Gumbel} - \text{Softmax}(\mathbf{s}). \quad (6)$$

We concatenate  $\mathbf{z}_g$  and  $\mathbf{z}_d$  along the feature dimension, then perform dimension reduction using a  $1 \times 1$  Convolution. This reduced-dimensional feature is subsequently element-wise multiplied by the broadcast binary mask  $\mathbf{M}$ . Thus, our path representation can be written as:

$$\mathbf{p} = \text{G}(\text{Conv}([\mathbf{z}_g, \mathbf{z}_d]) * \mathbf{M}), \quad (7)$$

$\text{G}(\cdot)$  represents the sorting of tokens. We place the masked tokens at the beginning as the starting point, while the remaining tokens follow their original order.

### 3.3 Path Signature Features

Considering the varying lengths and speeds of gestures, the application of Path Signature (PS) enables effective filtering of these variations and introduces non-linearity [31], leading to robust and versatile recognition capabilities. More details about path signature can be found in [16, 40].

Path signature is composed of path integrals. Suppose a  $D$ -dimensional path  $\mathbf{P} : [0, T] \rightarrow R^D$ , the coordinates of  $\mathbf{P}$  at time  $\tau \in [0, T]$  can be written as  $\mathbf{P}_\tau = (\mathbf{P}_\tau^1, \mathbf{P}_\tau^2, \dots, \mathbf{P}_\tau^D)$ . Path signature is a graded infinite series. To ensure the dimension of the path signature in a reasonable range, we usually consider the signature truncated at a certain level  $n$ :

$$\text{Sig}_n(\mathbf{P}) = (1, SN_1(\mathbf{P})_{0,T}, SN_2(\mathbf{P})_{0,T}, \dots, SN_n(\mathbf{P})_{0,T}) \quad (8)$$

The  $0^{th}$  term (i.e. a constant value set to 1) is optional for feature set.  $SN_n(\mathbf{P})_{0,T}$  means the  $n^{th}$  fold iterated integrals, which have many algebraic and analytic properties. For example, the  $1^{st}$  fold integral specifically signifies the increment in dimension.

For discrete skeleton sequence, with linear interpolation, the signature of each line segment of  $\mathbf{p}$  can be written as:

$$SN(\mathbf{p})_{\tau, \tau+1}^{d_i, \dots, d_n} = \frac{1}{n!} \prod_{j=1}^n (\mathbf{p}_{\tau+1}^{d_j} - \mathbf{p}_{\tau}^{d_j}). \quad (9)$$

Moreover, the entire  $\mathbf{p}$  can be computed according to Chen’s identity[3] state and Eq.(9).

## 4 Experiments

### 4.1 Datasets

We evaluated our method on two mainstream datasets related to gestures.

**ChaLearn2013** multimodal gesture dataset[12], which provides RGB, depth, foreground segmentation and skeleton data, contains 20 Italian gestures performed by 27 different persons. Each sequence lasts 1-2 minutes and includes 8-20 gesture instances. This dataset is split into training, validation and testing sets, containing 6850, 3454 and 3579 samples respectively. We only use skeleton data contains 19 joints for gesture recognition.

**WLASL(WLASL – 300subset)** The Word-Level American Sign Language dataset [25] contains 2,000 distinct ASL signs performed by more than 1,000 signers and was captured using RGB-D cameras, providing both color and depth data. The dataset has been divided into four subsets: WLASL-100, WLASL-300, WLASL-1000, and WLASL-2000. The WLASL-300 subset contains 5,117 videos of 300 different sign language classes, performed by 109 signers. In our experiments, we used skeleton data obtained from [25], which includes 55 joints for the body, left and right hand.

### 4.2 Implementation Details

We use cross-entropy as the loss, for WLASL Dataset, we add label smoothing with a weight of 0.1. SGD with momentum is used as the optimizer. The learning rate is updated between 1e-7 to 1e-2 with a step of 1060. The weight decay coefficient is 1e-5, and the batch size is 64 in ChaLearn2013 dataset and 32 in WLASL datasets. The network is trained on two GeForce GTX 1080 GPUs using PyTorch.

### 4.3 Ablation Study

*(i) Effectiveness* We verified the effectiveness of Motion-Guided Cluster Attention Block (MGCAB) and Path Signature (PS) on the ChaLearn2013 dataset, and the results are presented in Table 1. Regarding MGCAB, we eliminated the MGCAB module and substituted it with self-attention layers of equal or greater depth. The table shows that increasing the number of self-attention layers does

**Table 1.** The effectiveness of Motion-Guided Cluster Attention Block and Path Signature

Methods		Acc (%)	
MGCAB		PS	
w/o	with SA	w/o	with FC
✓			92.37
	✓		93.01
		✓	92.23
			93.27
<b>proposed</b>		<b>95.18</b>	

not significantly improve performance. As the number of self-attention layers grows, token similarity increases, causing the network to prefer retaining half of the tokens since it cannot identify distinctive tokens. Concerning the PS module, we replaced it with a FC layer or removed PS. As shown in the table, PS plays an essential role in comprehending the generated paths effectively.

*(ii) Network Structure* We evaluated various combinations of dynamic and global modules on different datasets and summarized the results in Table 2. The number of clusters are obtained by multiplying the number of tokens by certain ratios which determined through experiments. In the case of the ChaLearn2013 dataset, the hand joints only constitute a small proportion among the 19 joints involved in gestures, whereas for the WLASL datasets, 42 of the 55 are hand joints. Due to the small hand region and the fine-grained nature of sign language actions, a different number of layers and clustering numbers are required. The table shows that for ChaLearn2013 dataset, adding dynamic information in moderation is beneficial for achieving high accuracy. However, when there is an excessive amount of dynamic information and insufficient global information, the accuracy decrease. Meanwhile, in comparison with WLASL dataset, we found that it is more sensitive to dynamic information, and simply adding global information layers has a detrimental effect.

**Table 2.** the Network structure

Datasets	Layers	Cluster Num.	Acc (%)
ChaLearn2013	[12]	[-]	92.37
	[3, 12]	[10, -]	93.78
	[3, 3, 12]	[10, 5, -]	94.65
	[3, 4, 3, 8]	[15, 10, 5, -]	92.29
WLASL-300	[3, 3, 6]	[12, 6, -]	58.60
	[3, 4, 3, 3]	[12, 6, 2, -]	60.85
	[4, 4, 4, 3]		61.93
	[3, 4, 3, 6]	[12, 6, 2, -]	59.16
	[2, 3, 4, 3, 3]	[18, 12, 6, 2, -]	59.61

**Table 3.** Classification accuracy comparison against state-of-the-art methods on the ChaLearn2013 dataset.

Methods	Params(M)	FLOPs(G)	Acc (%)
PT-Logsig-RNN [31]	13.0	-	92.86
Two-stream LSTM [45]	-	-	91.70
CNN for Skeleton [10]	-	-	91.20
3s_net_TTM [24]	-	-	92.80
Multi-path CNN [30]	-	-	93.13
ST-PSM+L-PSM [5]	1.3	0.02	94.18
Shift-GCN [7]	0.6	-	90.86
AS-GCN [41]	6.9	0.54	92.66
CTR-GCN [4]	1.4	0.24	92.82
ST-GCN [48]	3.1	0.51	93.11
HD-GCN [22]	1.5	0.85	93.27
MS-G3D [34]	4.6	0.54	94.71
STFFormer [38]	5.5	2.44	92.77
ST-TR [37]	19.4	3.37	93.50
<b>Proposed</b>	<b>1.1</b>	<b>0.20</b>	<b>95.18</b>

#### 4.4 Comparison to the State-of-the-Art

We compared the results of some advanced work on two datasets, as shown in Table 3, 4. Among them, the methods in the first three parts of Table 3 are developed based on RNN, CNN, and GCN, while the fourth part combines Transformer related architectures. [5, 24, 31] also use the Path Signature.

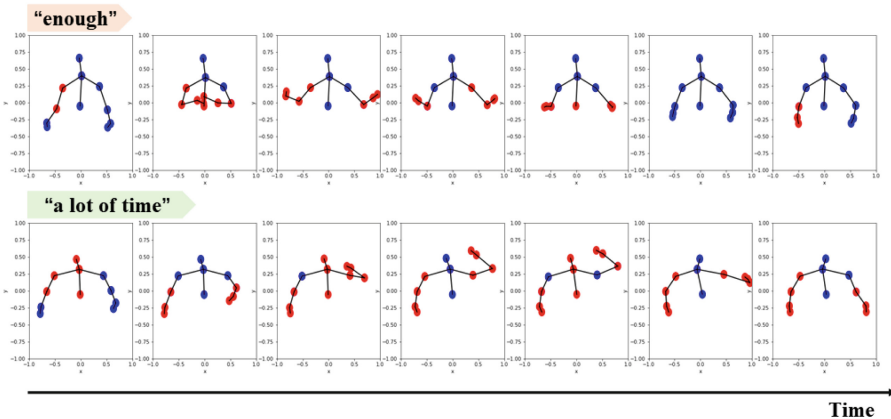
Specifically, [5] defines spatio-temporal paths and learnable paths, each composed of 2 or 3 joints, emphasizing local features of gesture execution. Our method extends the definition of paths by connecting multiple different joints across the global spatio-temporal domain. The global representation captures the dynamic changes and spatio-temporal relationships throughout the entire gesture execution process, thereby better adapting to different gesture variations. [37, 38] also incorporate global information. However, [37] models in the spatio-temporal domain separately, and [38] establishes local spatio-temporal modeling before linking these local spatio-temporal associations, which increases the computational complexity of the model. In terms of model parameters, our approach demonstrates a certain advantage over [5, 37, 38].

By utilizing path signatures, our model learns more discriminative gesture representations with fewer parameters. However, the dimensionality of global paths is larger than that of local paths, leading to increased FLOPs when calculating path signature features. When handling complex gestures, the computational complexity and resource requirements of the model can increase significantly. It is worth mentioning that [34] and we have achieved similar results, but its parameter quantity is nearly twice that of us. Therefore, combining effectiveness and efficiency, our method performs the best.

**Table 4.** Classification accuracy comparison against state-of-the-art methods on the WLASL-300 dataset.

Methods	Data	Params	Acc (%)
I3D [25]	RGB	12.35M	56.14
TK-3D [26]		-	68.75
Fusion-3 [19]		-	68.30
Pose-TGCN [25]	Skeleton	-	38.32
Pose-GRU [25]		-	33.68
GCN-Bert [43]		-	42.18
SPOTER [1]		5.92M	43.78
P3D(2D) [23]		4.94M	52.17
<b>Proposed</b>	Skeleton	1.25M	<b>60.85</b>

In the comparison of WLASL-300, there is still a significant gap between skeleton based methods and RGB based methods overall. But our method has improved by 4% compared to I3D [25], with only 1.25 million parameters compared to I3D’s 12.35 million. When utilizing the 2D skeleton as input, our method has a more significant improvement compared to [23] with a much smaller model size, indicating that path representation is indeed a concise and effective signature for gestures. However, the accuracy of RGB-based and skeleton-based methods is generally low. Capturing fine-grained motion patterns of identical gestures continues to be a significant challenge, particularly in scenarios with limited data (Fig. 4).



**Fig. 4.** The key joints selected (red) in the skeleton sequence.

## 4.5 Visualization

We conducted visualization on the ChaLearn2013, where we visualized two gesture sequences, “enough” and “a lot of time”, respectively. The red circles represent the points selected by the network to form the path. From the visualization, we can see that in the spatial domain, the network mainly focuses on the hands, elbows and shoulder. In some frames, none of the joints were selected, indicating that the network avoided repeating or unimportant information in the temporal dimension. However, the key joints of the hands are relatively scarce in ChaLearn2013 dataset, which might be the reason for the smaller accuracy improvement comparing with WLASL dataset.

## 5 Conclusion

In this work, we construct the Adaptive Global Gesture Path and introduce Path Signature, a powerful mathematical tool to help the network adapt to the temporal dynamics of gestures. In order to adaptively preserve the informative tokens of the entire gesture recognition, we synergize the global and dynamic branches to compute probability scores for token retention. The proposed Motion-Guided Cluster Attention Block analyzes the motion trajectories contained in tokens and aggregate relevant dynamic information. Based on the above ideas, we have constructed a powerful gesture recognition framework. The experiment proves that our proposed method is optimal in terms of both accuracy and parameter quantity. In addition, ablation experiments demonstrated the effectiveness of our proposed modules.

**Acknowledgements.** This work is supported by Key-Area Research and Development Program of Guangdong Province(2023B0303040001), Guangdong Basic and Applied Basic Research Foundation(2024A1515010180) and Guangdong Provincial Key Laboratory of Human Digital Twin(2022B1212010004). The work of Hao Ni is supported by the EPSRC under the program grant(EP/S026347/1) and the Alan Turing Institute under the EPSRC grant(EP/N510129/1).

## References

1. Boháček, M., Hrz, M.: Sign pose-based transformer for word-level sign language recognition. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 182–191 (2022)
2. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 172–186 (2018), <https://api.semanticscholar.org/CorpusID:198169848>
3. Chen, K.T.: Integration of paths-a faithful representation of paths by noncommutative formal power series. *Trans. Am. Math. Soc.* **89**(2), 395–407 (1958)
4. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13359–13368 (2021)

5. Cheng, J., Shi, D., Li, C., Li, Y., Ni, H., Jin, L., Zhang, X.: Skeleton-based gesture recognition with learnable paths and signature features. *IEEE Trans. Multimedia* **26**, 3951–3961 (2024). <https://doi.org/10.1109/TMM.2023.3318242>
6. Cheng, J., Zhang, X., Ni, H., Li, C., Xu, X., Wu, Z., Wang, L., Lin, W., Li, G.: Path signature neural network of cortical features for prediction of infant cognitive scores. *IEEE Trans. Med. Imaging* **41**(7), 1665–1676 (2022)
7. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 183–192 (2020)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021), <https://openreview.net/forum?id=YicbFdNTTy>
9. Du, M., Ding, S., Jia, H.: Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowl.-Based Syst.* **99**, 135–145 (2016)
10. Du, Y., Fu, Y., Wang, L.: Skeleton based action recognition with convolutional neural network. In: *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. pp. 579–583. *IEEE* (2015)
11. Duan, H., Zhao, Y., Chen, K., Lin, D., Dai, B.: Revisiting skeleton-based action recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 2969–2978 (2022)
12. Escalera, S., González, J., Baró, X., Reyes, M., Lopes, O., Guyon, I., Athitsos, V., Escalante, H.: Multi-modal gesture recognition challenge 2013: Dataset and results. In: *Proceedings of the 15th ACM on International conference on multi-modal interaction*. pp. 445–452 (2013)
13. Fayyaz, M., Koochpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: *European Conference on Computer Vision*. pp. 396–414. *Springer* (2022)
14. Ge, L., Cai, Y., Weng, J., Yuan, J.: Hand pointnet: 3d hand pose estimation using point sets. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* pp. 8417–8426 (2018), <https://api.semanticscholar.org/CorpusID:52045802>
15. Gong, C., Wang, D., Li, M., Chandra, V., Liu, Q.: Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753* (2021)
16. Graham, B.: Sparse arrays of signatures for online character recognition. *arXiv preprint arXiv:1308.0371* (2013)
17. Hambly, B., Lyons, T.: Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics* pp. 109–167 (2010)
18. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 16000–16009 (2022)
19. Hosain, A.A., Santhalingam, P.S., Pathak, P., Rangwala, H., Kosecka, J.: Hand pose guided 3d pooling for word-level sign language recognition. In: *WACV*. pp. 3429–3439 (2021)
20. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016)

21. Lai, S., Zhu, Y., Jin, L.: Encoding pathlet and sift features with bagged vlad for historical writer identification. *IEEE Trans. Inf. Forensics Secur.* **15**, 3553–3566 (2020)
22. Lee, J., Lee, M., Lee, D., Lee, S.: Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 10444–10453 (October 2023)
23. Lee, T., Oh, Y., Lee, K.M.: Human part-wise 3d motion context learning for sign language recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 20740–20750 (2023)
24. Li, C., Zhang, X., Liao, L., Jin, L., Yang, W.: Skeleton-based gesture recognition using several fully connected layers with path signature features and temporal transformer module. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 33, pp. 8585–8593 (2019)
25. Li, D., Rodriguez, C., Yu, X., Li, H.: Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In: *WACV*. pp. 1459–1469 (2020)
26. Li, D., Yu, X., Xu, C., Petersson, L., Li, H.: Transferring cross-domain knowledge for video sign language recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 6205–6214 (2020)
27. Li, L., Wang, M., Ni, B., Wang, H., Yang, J., Zhang, W.: 3d human action representation learning via cross-view consistency pursuit. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 4741–4750 (2021)
28. Li, Y., Ma, D., Yu, Y., Wei, G., Zhou, Y.: Compact joints encoding for skeleton-based dynamic hand gesture recognition. *Computers & Graphics* **97**, 191–199 (2021)
29. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. In: *International Conference on Learning Representations* (2022)
30. Liao, L., Zhang, X., Li, C.: Multi-path convolutional neural network based on rectangular kernel with path signature features for gesture recognition. In: *2019 IEEE Visual Communications and Image Processing*. pp. 1–4. IEEE (2019)
31. Liao, S., Lyons, T., Yang, W., Schlegel, K., Ni, H.: Logsig-rnn: a novel network for robust and efficient skeleton-based action recognition. *arXiv preprint arXiv:2110.13008* (2021)
32. Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S., Pan, C.: Decoupled representation learning for skeleton-based gesture recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5751–5760 (2020)
33. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 10012–10022 (2021)
34. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 143–152 (2020)
35. Lyons, T., Ni, H., Oberhauser, H.: A feature set for streams and an application to high-frequency financial tick data. In: *Proceedings of the 2014 International Conference on Big Data Science and Computing*. pp. 1–8 (2014)



36. Mou, C., Zhang, X.: Attention based dual branches fingertip detection network and virtual key system. Proceedings of the 28th ACM International Conference on Multimedia (2020), <https://api.semanticscholar.org/CorpusID:222277881>
37. Plizzari, C., Cannici, M., Matteucci, M.: Skeleton-based action recognition via spatial and temporal transformer networks. *CVIU* **208**, 103219 (2021)
38. Qiu, H., Hou, B., Ren, B., Zhang, X.: Spatio-temporal tuples transformer for skeleton-based action recognition. arXiv preprint [arXiv:2201.02849](https://arxiv.org/abs/2201.02849) (2022)
39. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Adv. Neural. Inf. Process. Syst.* **34**, 13937–13949 (2021)
40. Reizenstein, J.F., Graham, B.: Algorithm 1004: The iisignature library: Efficient calculation of iterated-integral signatures and log signatures. *ACM Transactions on Mathematical Software* **46**(1), 1–21 (2020)
41. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12026–12035 (2019)
42. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In: Proceedings of the Asian conference on computer vision (2020)
43. Tunga, A., Nuthalapati, S.V., Wachs, J.: Pose-based sign language recognition using gcn and bert. In: WACV. pp. 31–40 (2021)
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
45. Wang, H., Wang, L.: Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 499–508 (2017)
46. Weng, J., Liu, M., Jiang, X., Yuan, J.: Deformable pose traversal convolution for 3d action and gesture recognition. In: European Conference on Computer Vision. pp. 136–152 (2018)
47. Xie, Z., Sun, Z., Jin, L., Ni, H., Lyons, T.: Learning spatial-semantic context with fully convolutional recurrent network for online handwritten chinese text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(8), 1903–1917 (2017)
48. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
49. Yang, W., Lyons, T., Ni, H., Schmid, C., Jin, L.: Developing the path signature methodology and its application to landmark-based human action recognition. In: *Stochastic Analysis, Filtering, and Stochastic Optimization: A Commemorative Volume to Honor Mark HA Davis's Contributions*, pp. 431–464. Springer (2022)
50. Yin, H., Vahdat, A., Alvarez, J., Mallya, A., Kautz, J., Molchanov, P.: A-ViT: Adaptive tokens for efficient vision transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2022)
51. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11101–11111 (2022)
52. Zhang, X., Cheng, J., Ni, H., Li, C., Xu, X., Wu, Z., Wang, L., Lin, W., Shen, D., Li, G.: Infant cognitive scores prediction with multi-stream attention-based temporal path signature features. In: *Medical Image Computing and Computer Assisted Intervention*. pp. 134–144. Springer (2020)



# Self-supervised Multi-actor Social Activity Understanding in Streaming Videos

Shubham Trehan and Sathyanarayanan N. Aakur<sup>(✉)</sup>

CSSE Department, Auburn University, Auburn, AL 36849, USA  
{szt0113,san0028}@auburn.edu

**Abstract.** This work addresses the problem of Social Activity Recognition (SAR), a critical component in real-world tasks like surveillance and assistive robotics. Unlike traditional event understanding approaches, SAR necessitates modeling individual actors' appearance and motions and contextualizing them within their social interactions. Traditional action localization methods fall short due to their single-actor, single-action assumption. Previous SAR research has relied heavily on densely annotated data, but privacy concerns limit their applicability in real-world settings. We propose a self-supervised approach based on multi-actor predictive learning for streaming SAR. Using a visual-semantic graph, we model social interactions, enabling relational reasoning for robust performance without labeled data. The proposed framework achieves competitive performance on standard group activity recognition benchmarks. Evaluation on three publicly available action localization benchmarks demonstrates its generalizability to arbitrary action localization.

**Keywords:** Group Activity Recognition · Action Localization

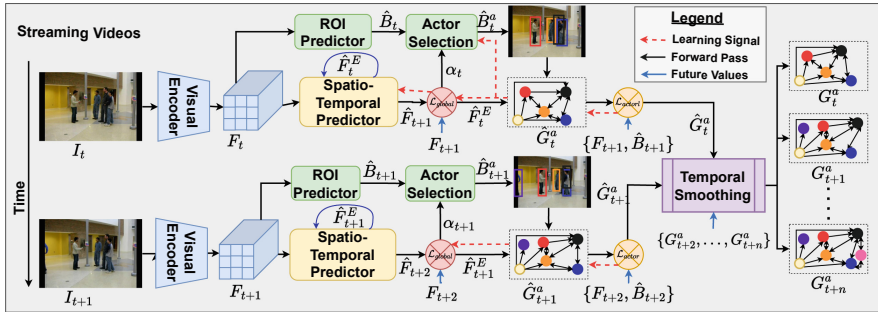
## 1 Introduction

Social activity recognition (SAR) is a key part of computer vision applications in the real world, such as surveillance and assistive robotic systems. It differs from traditional event understanding approaches [1, 2, 8, 35, 36] since it requires the modeling of individual actor's appearance and their motions, and contextualizing them within the scope of their social interactions. SAR brings a unique set of challenges. First, there is a need for actor localization, social relationship modeling, and social activity recognition. Second, the number of actors in each frame can change due to occlusion, camera range, or noise due to missed/false detection. Finally, a scene can have an arbitrary number of social groups. Traditional action localization approaches [1, 2, 35] cannot be directly extended to this problem since they assume a single action performed by a single actor.

The dominant approach has been to learn the social dynamics of a scene using attention-based or graph-based relational reasoning in a supervised learning setting. The key assumption has been the availability of densely annotated data for

training and near-perfect actor localization. Hence, the literature has focused on feature aggregation across time and social groups. While this has yielded tremendous progress, it is not always possible to expect densely annotated data for training, primarily due to the privacy concerns involved in collecting, storing, and annotating visual data in a social setting. There is a need to move away from over-reliance on labeled training data and towards self-supervised learning approaches that can learn in an open world, i.e., unconstrained training and test semantics, and a streaming fashion, i.e., learning with a single pass through the data without storing it without loss of generalization.

In this work, we focus on addressing social activity understanding in streaming videos without labeled data. We propose multi-actor predictive learning for jointly modeling actor-level actions and contextual, group-level activities. We move away from the single-actor, single-action assumption from prior approaches [1–3, 42] and propose to represent visual scenes in a social-contextualized action graph for social event understanding. The **contributions** of our approach are four-fold: (i) we are the first to tackle the problem of self-supervised social activity detection in streaming videos, (ii) we introduce a visual-semantic graph structure called an *action graph* to model the social interaction between actors in a group setting, (iii) we show that relational reasoning over this graph structure by spatial and temporal graph smoothing can help learn the social structure of cluttered scenes in a self-supervised manner requiring only a single pass through the training data to achieve robust performance, and (iv) we show that the framework can generalize to arbitrary action localization without bells and whistles to achieve competitive performance on publicly available benchmarks.



**Fig. 1. Overall architecture.** Using multi-actor predictive learning, we can localize actors and model their interactions as an action graph, which can be used for downstream event understanding tasks such as action and social activity detection.

## 2 Related Work

**Group activity recognition** has been a widely studied area of social event understanding. The typical pipeline starts with actor detection, individual fea-

ture extraction, and social interaction modeling. Action features are extracted for each actor using pre-trained action recognition models [8]. The primary mechanism has been to model the social interaction of actors within a group setting using attention-based mechanisms to generate social group features [37], model individual actor dynamics [11], to model the spatial and temporal dependencies [22] jointly, or for multi-view representation learning [32]. Others use transformers [43] to bypass object detection requirements [18], model keypoint dynamics [50], social relation modeling [10], and spatiotemporal multiscale feature aggregation [53] to reduce training requirements. Note that group activity recognition is a special case of social activity recognition (SAR), where the underlying assumption is that all actors are part of a single social group that works together to perform a collective action. SAR does not make any such assumptions about the social structure.

**Relational reasoning** is another line of work that focuses on aggregating actor-actor interactions for group activity understanding. These approaches aim to model the spatiotemporal dependencies by considering the spatial relationships between objects using a variety of mechanisms such as aggregating the relational contexts and scene information using transformers [29], using graph convolutional networks (GCNs) to capture the appearance and position relation between actors [14, 45], or capture spatial coherence using recurrent neural networks (RNNs) [31, 44], convolutional neural networks (CNNs) [4], graph-LSTMs [34, 38], graph attention [27], factor graphs [47], knowledge distillation [39], tokenization [46], tracking [40, 41], and contrastive learning [12], to name a few. The prevalent paradigm in the above approaches has been supervised learning to establish and learn social interactions, with varying levels of supervision, i.e., bounding box locations and labels of individual actors, group activity labels, and social group memberships, which requires immense human effort for annotations and may reduce their generalizability. Some works [6, 18, 49, 50, 53] have attempted to reduce the training requirements by relaxing assumptions about annotation granularity but still require large amounts of data.

Our work is one of the first to tackle this problem from a self-supervised learning perspective by modeling the actor dynamics from a multi-actor predictive learning perspective. *Predictive learning* has emerged as a powerful paradigm for visual event understanding. Proposed in cognitive science literature [51], the goal of predictive learning is to learn representations by anticipating the future and use the residuals for downstream tasks such as event segmentation [3], action localization [1, 2], active object tracking [42], future frame generation [26], and hierarchical event perception [28], among others. All prior works have focused on single-actor settings, where only one global action is expected to be present. We offer a unique perspective on predictive learning by extending the idea to multi-actor predictions for group activity detection. We do not require annotations and aim to learn robust representations at both the actor level and group level, while feature aggregation allows us to model social interactions.

### 3 Proposed Framework

**Overview.** We propose to tackle the problem of multi-actor, multi-action localization in streaming videos. The overall framework is illustrated in Figure 1. Given a sequence (stream) of video frames  $\{I_0, I_1, \dots, I_t\}$ , we aim to localize actors of interests, characterized by their location (bounding boxes)  $\hat{B}_i^a$  and visual features  $F_t^a$ . We then construct a graph structure called an *action graph* ( $\hat{G}_t^a$ ) whose nodes are actors and edges are social interactions, along with an event node. The event and action level features are contextualized using a temporal smoothing layer to construct a final action graph that can be used for various downstream tasks such as group activity understanding, social activity understanding, and arbitrary action localization.

#### 3.1 Visual Perception and ROI Prediction

Our framework begins with a visual perception module, aiming to extract visual features at both scene and object levels. Our primary visual perception module uses a DETR [7] model. For every frame  $I_t$ , we extract (i) global scene features,  $F_t$ , (ii) object regions of interests (ROI),  $\hat{B}_t$ , and (iii) object-level features,  $F_t^B$ . The ResNet backbone provides a lower-resolution, global feature map  $F_t \in \mathbb{R}^{2048 \times H \times W}$ , where  $W$  and  $H$  are the spatial resolution of the global feature map. DETR’s detection heads are used to generate initial object ROI proposals  $\hat{B}_t$ , i.e., the search space for actor localization, and the decoder outputs for each ROI prediction are used for object-level features ( $F_t^B$ ). Note that at this stage, we only generate actor candidates that will be refined using the actor selection module described in Section 3.2. We do not fine-tune DETR on the video datasets and use a model pre-trained on MS-COCO [24]. During training, all objects are considered as candidate actors in a class-agnostic manner, following prior works [1, 2], while we filter out only “human” predictions during inference.

#### 3.2 Spatiotemporal Prediction for Actor Localization

We model the spatial-temporal dynamics of the scene using a spatiotemporal predictor module. The goal is to learn an event-level representation ( $\hat{F}_t^E$ ) that captures how each object, represented by its location  $B_t$  and visual features  $F_t^B$ , changes over time. We use a simple  $L$ -layer LSTM stack as our spatiotemporal predictor, which takes the global scene-level feature  $F_t$  as input and anticipates the future global representation  $F_{t+1}$ . The goal is not to predict the future frame by pixel-level regression but to model how the scene changes over time. This event representation  $\hat{F}_t^E$  is continuously updated as new frames ( $I_t$ ) are observed in a streaming fashion using a predictive loss function given by

$$\mathcal{L}_{global} = \frac{1}{H * W} \sum_{H,W} M_t \odot \|F_{t+1} - \hat{F}_{t+1}\|_2 \quad (1)$$

where  $M_t$  is the motion difference between frames  $I_t$  and  $I_{t+1}$  computed as the first-order hold between  $F_t$  and  $F_{t+1}$ ;  $\hat{F}_{t+1}$  is the anticipated global feature at

time  $t+1$ , obtained by projecting the event feature  $\hat{F}_t^E$  back to the 2-D feature space. The predictive loss enables the LSTM stack to learn a robust event representation ( $\hat{F}_t^E$ ) that can anticipate the future scene’s spatial features  $F_{t+1}$ .

**Actor Selection** The unnormalized prediction errors ( $P_t = M_t \odot \|F_{t+1} - \hat{F}_{t+1}\|_2$ ) from Equation 1 are proportional to the predictability of each spatial location. Hence, higher prediction errors indicate the presence of a less predictable foreground action(s), while lower prediction errors indicate a more predictable background action. We formulate a prediction-driven attention mask  $\alpha_t$  by passing  $P_t$  through a softmax activation function to increase focus on foreground actions while suppressing background actions. The top  $K$  attention “slots” are used to filter object ROIs  $\hat{B}_t$  and select the *actor* ROIs  $\hat{B}_t^a$ . Note that actor ROIs  $\hat{B}_t^a$  are predicted only if the prediction-based attention slots  $\alpha_t^{ij}$  fall within any ROI  $b_t^k \in \hat{B}_t$ . Hence, the number of actors chosen from candidate ROIs is much lower, allowing us to model actor-level dynamics better.

**Building an Action Graph** We construct a graph  $\hat{G}_t^a$  for every observed frame  $I_t$ , with actors as nodes  $\mathcal{V}_t$  to model actor-level interaction dynamics. Each node  $N_i \in \hat{G}_t^a$  is described by a feature vector  ${}_i\hat{F}_t^a = [{}_iF_t^B; {}_i\hat{B}_t^a]$  that captures its geometry and visual features. The edges in this graph structure,  $\mathcal{E}_t$ , are defined by the spatial structure of the actors selected using the prediction-based attention  $\alpha_t$ . Unlike previous graph-based approaches [10, 45], we do not use a fully connected structure. Instead, we model their social connectivity using a distance-based formulation. An adjacency matrix  $A_t$  is constructed by computing the spatial proximity between each pair of nodes, given by the Euclidean distance  $\phi$  between their locations and spatial geometry, and centering it by subtracting the mean distance between all nodes. The adjacency for each node  $N_i$  is normalized as  $A_t^i = \sigma(\frac{A_t^i}{\|A_t^i\|_2})$ , to ensure that the distances are scaled proportionally and  $\sigma$  is the Sigmoid function. The adjacency matrix is thresholded to get the final social structure by discarding all edges less than the average normalized distance in the adjacency. This formulation allows us to model the social interactions between the actors detected in the scene without the underlying assumption that all actors interact with each other, regardless of their social activity. Finally, an “action node”, instantiated by the event features  $\hat{F}_t^E$ , is added to the graph and is connected to all actor nodes. This additional node allows us to propagate action features to relevant actors, and the connections between actor nodes will enable us to capture contextual cues for modeling actions with interacting actors, as described in the next section. Empirically, in Section 4, we see that adding the action node and the subsequent contextualization using graph and temporal smoothing plays a big role in improving the performance of both group activity recognition and individual activity detection. The action graph formulation distinguishes us from prior unsupervised event understanding approaches [1–3] since it allows us to model each actor individually without any prior assumptions about their role or interactions in a social group setting.

### 3.3 Contextualizing Cues with Graph and Temporal Smoothing

Recognizing social activity and individual actions in a group setting requires reasoning over the spatial interaction between actors at every instant and its evolution over time. To this end, given our action graph  $\hat{G}_t^a$ , the next step is contextualizing each person’s action using a two-step spatial-temporal graph smoothing process. First, we use a message passing layer, as introduced in Graph Convolutional Networks (GCNs) [45], for spatial reasoning over the social interaction between actors as captured in  $\hat{G}_t^a$ . Formally, this is defined as

$$F_t^a = \sigma(A_t \hat{F}_t^a W_s) \quad (2)$$

where  $A_t$  is the adjacency matrix for  $G_t^a$ ,  $\hat{F}_t^a$  is the feature representation for each node of the action graph,  $W_s$  is the learnable parameter matrix for the GCN layer, and  $\sigma$  is the ReLU activation function. The resulting features  $F_t^a$  are contextualized across actors, conditioned on their social structure (specified by weighted edges  $\mathcal{E}_t$ ), and the event-level features  $\hat{F}_t^E$  represented by the action node in  $G_t^a$ . While this reasoning layer can be repeated, additional layers harm the model’s performance (see Section 4) due to the homogenization of features.

For temporal contextualization, we construct a composite spatial-temporal graph by establishing temporal edges between actor nodes in  $G_t^a$  with their corresponding nodes in the subsequent graph  $G_{t+1}^a$ . While straightforward in theory, we must address two critical challenges for implementation. First, we do not have the ground truth tracking annotations that would enable us to establish actor-actor correspondences across frames. Second, the number of detected actors is not constant across time, requiring comparing graphs of different sizes. Hence, registering nodes across actor graphs between consecutive frames requires us (i) to establish a permutation matrix  $\mathcal{P}$  to account for varying node ordering across graphs and (ii) to add null nodes (representing missed/false detections) to the graph with fewer nodes to ensure every node is registered to one node across time. The optimal permutation matrix  $\mathcal{P}$  is obtained by computing a one-to-one match between two graphs  $G_t^a$  and  $G_{t+1}^a$  using the Hungarian matching algorithm to minimize the distance between the two graphs. Formally, this is the optimization for

$$\arg \min_{\mathcal{P} \in \mathcal{P}_n} \sum_{i=1}^N w_1 \|iF_t^a - \mathcal{P}(iF_{t+1}^a)\|_2 + w_2 \phi(iB_t^a - \mathcal{P}(iB_{t+1}^a)) \quad (3)$$

where  $N$  is the total number of nodes in the graphs  $G_t^a$  and  $G_{t+1}^a$ ,  $\mathcal{P}_n$  is the space over all permutation matrices,  $\phi$  is the Intersection over Union (IoU) distance between two bounding boxes, the function  $\mathcal{P}(\cdot)$  results in the transformation of a given set of nodes after applying a permutation matrix, i.e.,  $v \mapsto \mathcal{P}v$ , and  $w_1$  and  $w_2$  are scaling factors to balance the two difference distances (i.e., between feature distance and IoU distance across nodes, respectively). Finally, based on this learned permutation matrix, we establish temporal edges between the nodes registered across time. The composite adjacency matrix  $\mathcal{A}_G$  and the corresponding action feature matrix  $\mathcal{F}_G$  are used to construct the spatial-temporal graph

( $\mathcal{G}_a$ ), representing the entire video  $\mathcal{V}=I_1, I_2, \dots, I_T$ . A temporal smoothing is performed on  $\mathcal{G}_a$  to get the final actor-level features, as defined by

$$\hat{\mathcal{F}}_G = \sigma(\mathcal{A}_G \mathcal{F}_G W_t) \quad (4)$$

where  $W_t$  is a fixed (non-learnable) identity matrix, making the operation purely based on message passing. Similar to the spatial smoothing process, this process can be repeated, but empirically, we find one layer is ideal for our experiments. As seen in Section 4, temporal smoothing provides substantial gains in group activity recognition and action detection.

### 3.4 Social Modeling with Multi-actor Predictive Learning

In addition to the global, event-level predictive learning introduced in Equation 1, we introduce the notion of multi-actor predictive learning. This allows us to model the spatial-temporal dynamics of all actors, conditioned on their social interactions and the overall event dynamics of the scene. We model this using a multi-actor prediction loss given by

$$\mathcal{L}_{actor} = \frac{1}{N} \sum_{i=0}^N \|\hat{F}_t^a - \mathcal{P}(i\hat{F}_{t+1}^a)\|_2 + \|B_t^a - \mathcal{P}(iB_{t+1}^a)\|_2 \quad (5)$$

where the first term minimizes the differences between the anticipated actor-level features and the actual actor-level features between consecutive frames, and the second minimizes their respective geometry. We anticipate the future feature and geometry of each actor using two fully connected neural networks defined by  $iF_{t+1}^a = W_{act} * iF_t^a$  and  $iB_{t+1}^a = W_{bb} * iF_t^a$ , respectively. This allows us to train our overall spatial-temporal prediction stack (defined in Equations 1 and 5) and the smoothing layers (Equations 2 and 4) by minimizing the overall prediction errors given by

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{global} + \lambda_2 \mathcal{L}_{actor} \quad (6)$$

where  $\lambda_1$  and  $\lambda_2$  allow us to balance the global event-level prediction loss and the actor-level multi-actor prediction loss.

**Inferring Labels.** For group activity recognition, we do mean average pooling over all actor-level features  $\hat{\mathcal{F}}_G$  defined in the composite spatial-temporal action graph  $\mathcal{G}_a$ . K-means clustering is performed on the mean-pooled features to obtain the final labels. K-means clustering over actor-level features  $\hat{\mathcal{F}}_G$  provides actor-level action labels. Following prior work [1–3], Hungarian matching is performed between the predicted labels and ground truth labels to compute the quantitative metrics, as defined in Section 4. A Spectral Clustering model is fit on the adjacency matrix  $\mathcal{A}_G$  to find social communities for social activity recognition, following the protocol from the prior work [10].

**Implementation Details** We use a DETR [7] model, pre-trained on MS COCO [24] as our ROI predictor. The CNN backbone is ResNet-50 [13]. The



**Table 1. Group Activity Recognition** results evaluated on the Collective Activities and Volleyball dataset. Accuracy is reported for group activity recognition (denoted as “Activity”) and mAP for individual action detection (denoted as “Action”). Note: “-” indicates the model does not *detect* individual actions.

Approach	Training Requirements			Bboxes for eval	CAD Dataset		Volleyball Dataset
	Bboxes	Ind. Labels	Grp.Labels		Activity	Action	Activity
HDTM[19]	✓	✓	✓	✓	81.5	-	81.9
HANs+HCNs[21]	✓	✓	✓	✓	84.3	-	85.1
CCGL[38]	✓	✓	✓	✓	90.0	-	87.6
CERN [33]	✓	✓	✓	✓	87.6	-	83.3
stagNet [30]	✓	✓	✓	✓	89.1	-	89.3
GAIM [20]	✓	✓	✓	✓	90.6	-	91.9
AT [11]	✓	✓	✓	✓	<u>90.8</u>	-	91.4
GroupFormer [22]	✓	✓	✓	✓	<b>93.6</b>	-	94.1
HIGCIN [48]	✓	✗	✓	✓	92.5	-	91.4
CRM [4]	✓	✓	✓	✗	83.4	-	92.1
SBGAR [23]	✗	✓	✓	✗	83.7	-	38.7
Zhang et al [52]	✓	✗	✓	✗	83.7	-	86.0
ARG [45]	✓	✓	✓	✗	86.10	49.60	-
Ehsanpour et. al.[10]	✓	✓	✓	✗	<u>89.40</u>	<u>55.90</u>	93.1
HGC-Former[37]	✓	✓	✓	✗	<b>96.50</b>	<b>64.90</b>	-
PredLearn( $K = K_{GT}$ )	✗	✗	✗	✗	62.83	-	10.05
AC-HPL( $K = K_{GT}$ )	✗	✗	✗	✗	<u>72.20</u>	-	24.58
Ours ( $K = K_{GT}$ )	✗	✗	✗	✗	<b>75.95</b>	<b>26.75</b>	<b>39.51</b>
Ours ( $K = K_{OPT}$ )	✗	✗	✗	✗	<b>90.41</b>	<b>33.02</b>	<b>43.28</b>

size of the global scene features  $F_t$  is  $7 \times 7 \times 2048$ . Class-agnostic detections with a confidence score of more than 0.1 are taken as object candidates during training. The top  $K = 25$  attention slots from the predictive learning error are used to select actors. We use a 2-layer LSTM network as our spatio-temporal predictor, defined in Section 3.2. The hidden size of each LSTM layer is set to 2048. The GCN layers for spatial and temporal smoothing are set to 512, and a fully connected layer is used to project the features back to 2048 for multi-actor predictive learning.  $w_1$  and  $w_2$  in Equation 3 are set to 1.  $\lambda_1$  and  $\lambda_2$  are set to 1 in Equation 6. As with PredLearn,  $K_{OPT}$  is set to be  $3 \times K_{GT}$ . The prediction stack’s learning rate is  $1 \times 10^{-4}$ , and for the spatial and temporal smoothing layers is  $1 \times 10^{-3}$ , found using a grid search between  $10^{-5}$  and  $10^{-2}$ . Training converges in 6 hours on a workstation with a 64-core AMD ThreadRipper, an RTX5500, and 128GB CPU RAM.

## 4 Experimental Evaluation

**Data.** We use the Collective Activities Dataset (CAD) [9], its annotations-augmented version, SocialCAD [10], and the Volleyball Dataset [15], to evaluate

our framework. CAD consists of 44 videos of people performing 6 individual actions across 5 group activities in unconstrained real-world scenarios. SocialCAD augmented CAD with additional information, such as individuals’ social group identification and collective social activity. We follow prior work [10] and use 31 videos for training and 11 for evaluation. The Volleyball Dataset [15] consists of 4,830 videos obtained from 55 volleyball matches, with 9 group actions annotated. Following prior work [37], we use 3,493 videos for training and 1,337 videos for evaluation. To evaluate the generalization capabilities of our framework to arbitrary action localization, we use three publicly available benchmarks - UCF Sports [36], JHMDB [16], and THUMOS’13 [17]. Each dataset contains a different number of actions (10 in UCF Sports, 21 in JHMDB, and 24 in THUMOS’13) across different domains (sports and daily activities). Each dataset offers a unique challenge for action localization, such as cluttered scenes, highly similar action classes, large camera motion, and object occlusion. We follow prior work [2, 35] and use official train-test splits for all datasets.

**Metrics.** We use different metrics for evaluating the performance on each task. We use the mean multi-class classification accuracy (MCA) for group activity recognition. For individual action detection, we follow prior work [10] and use the mean average precision (mAP) as the evaluation metric to account for missed and false detections. To evaluate social activity understanding, we use two different metrics - membership accuracy and social activity recognition, as defined in SocialCAD. The former measures the accuracy of recognizing a person’s social group in the video. The latter measures the ability to jointly predict a person’s membership and the social activity label. We report the video-level mAP at 0.5 IOU threshold for arbitrary action localization.

**Baselines.** We compare against various supervised, weakly supervised, and unsupervised learning approaches for both group activity understanding and action localization. The supervised [11, 19–22, 30, 33, 38] and weakly supervised learning baselines [4, 10, 23, 37, 45, 48] provide solid baselines for comparing the representation learning capabilities of our framework. Unsupervised learning approaches, particularly closely related approaches such as AC-HPL [2] and PredLearn [1], allow us to benchmark our approach with others trained under the same settings. We use Hungarian matching for all unsupervised learning baselines to align their predictions with the ground truth labels, following prior work [1, 2]. Note that all baselines, except AC-HPL and PredLearn, are not trained in a streaming fashion and require strong visual encoders pre-trained on large amounts of video data (e.g., I3D [8] on Kinetics [8]) and fine-tuned for a large number of epochs ( $> 50$ ). We do not require either and only use DETR [7] pre-trained on MS-COCO for person detection and train in a streaming fashion, requiring only one pass through all the videos.

## 4.1 Group Activity Recognition

We first evaluate our approach on the group activity *recognition* task, where the goal is to identify the activity in which the *majority* of the people are involved. Table 1 summarizes the results. As can be seen, we perform competitively with

**Table 2. Social Activity Understanding** results on the SocialCAD dataset [10]. Note: All results are in the detection setting, i.e., without GT bounding boxes.

Approach	Training Requirements			Membership Recognition	Social Activity Recognition
	Bboxes	Labels	Member		
GT [Group] (Upper Bound)	-	-	-	54.4	51.6
GT [Individual] (Upper Bound)	-	-	-	62.5	54.9
HGC-Former [37]	✓	✓	✓	-	46.0
ARG [Group] [45]	✓	✓	✓	49.0	34.8
Ehsanpour et al [Group] [10]	✓	✓	✗	49.0	35.6
Ehsanpour et al [individual] [10]	✓	✓	✗	52.4	41.8
Ours	✗	✗	✗	<b>32.33</b>	<b>25.07</b>

supervised learning approaches and significantly outperform prior unsupervised learning approaches such as PredLearn [1] (by 13.12%) and AC-HPL [2] (by 3.75%). We observe that some activity classes, such as “walking” and “crossing”, exhibit high intra-class variation in the clustering. Hence, we increase the number of clusters for recognition to its optimal number (using the elbow method with intra-cluster variation as the metric) and devise a baseline indicated by  $K = K_{OPT}$ . We observe that the accuracy increases significantly to 90.41%, outperforming many of the supervised and weakly supervised approaches. It is to be noted that the supervised learning approaches (at the top of Table 1) require ground truth bounding boxes during inference for efficient recognition. Weakly supervised approaches [10, 37, 45] do not require bounding boxes during inference but require supervision from dense annotations. Interestingly, we observe that the mean per-class accuracy (MPCA) is 81.25%, with the class “Waiting” being the worst-performing one at 35.51%. We attribute it to the predictive learning paradigm, which naturally focuses on actors with the least predictive motions. It has actors with highly predictable motion, which reduces the model’s attention on them and leads to poorer recognition accuracy. However, other classes have a recognition accuracy above 90%, indicating the model’s effectiveness in recognizing actions that involve reasoning over actor appearance and motion. On the Volleyball dataset, we obtain a group activity recognition accuracy of 39.51%, which significantly outperforms the other unsupervised baselines PredLearn (10.05%) and AC-HPL (24.58%). However, we observe a higher gap between the unsupervised and supervised models. We attribute it to the fact that the supervised baselines use an I-3D network pre-trained on large datasets such as Kinetics, are trained for over 50 epochs, and require densely annotated data (such as ground truth position information for individual players). Our model does not need such training requirements and can be trained in a streaming manner. Better modeling of social interactions and fine-grained visual feature integration can help improve performance and narrow the gap between supervised and unsupervised models.

In addition to group activity recognition, we also report the mAP score for individual action detection (last column of Table 1), where the goal is to localize and recognize every actor’s actions. As can be seen, we once again outperform

**Table 3. Generalization to arbitrary action localization.** We report the video-mAP and compare it with unsupervised *action* localization baselines. OOD refers to the evaluation on data other than the training domain (CAD).

Approach	OOD Eval	UCF Sport	JHMDB	THUMOS13
Ours	✓	<b>0.40</b>	<b>0.22</b>	<b>0.15</b>
AC-HPL[2]	✗	<b>0.59</b>	0.15	<u>0.20</u>
PredLearn[1]	✗	0.32	0.10	0.10
Soomro[35]	✗	0.30	<u>0.22</u>	0.06
Ours	✗	<u>0.49</u>	<b>0.25</b>	<b>0.21</b>

prior unsupervised learning approaches significantly while offering competitive performance to supervised learning approaches [10, 37, 45]. We do not finetune our ROI prediction (DETR) on the CAD dataset as with the supervised learning approaches. This significantly increases the number of actors detected in the scene, which is not always reflected in the ground truth. One such instance is highlighted in Figure 2, where we correctly localize and recognize the individual actions of *all* actors in the scene and not just those in the ground truth. Prior unsupervised learning approaches (PredLearn and AC-HPL) do not predict distinct actions for each actor, but rather, a collective group activity is assigned to each person. This reduces their utility in action detection and stems from their inherent assumption that there is one action per video and that all actors participate in this global action. We do not make such assumptions and can effectively recognize and localize multiple, simultaneous actions.

## 4.2 Social Activity Understanding

In addition to group activity recognition and individual action detection, we also evaluate the representation learning capabilities of our framework for social activity understanding tasks such as membership recognition and social activity recognition. For membership recognition, we follow Ehsanpour *et al.* [10] and use graph spectral clustering to segment the individual actors into social groups to compute the membership recognition accuracy. Table 2 summarizes the results. We perform competitively with supervised learning approaches such as HGC-Former [37] and ARG [45], which require prior knowledge of memberships during training. We also perform competitively with Ehsanpour *et al.* [10], which does not require membership labels during training but does require other annotations, such as individual and group labels, along with their bounding box annotations during training. The baselines GT[Group], taken from Ehsanpour *et al.* [10], provides the upper bound for detection-based models when the member locations and actions are provided, and an I3D model [8] is used for labeling the membership and social activity of the person. On inspecting the results, we find that much of the reduction in membership recognition accuracy is because we predict and localize more actions than provided in the ground truth and, hence,

make more predictions per frame. For example, in Figure 2, we detect the membership and actions of all people, not just those in the annotations. Fine-tuning DETR on the ground truth annotations will reduce the false alarms and improve the performance, albeit at the cost of generalization.

**Table 4. Ablation study** results on the collective activities dataset. We report accuracy for group activity recognition and the mAP for individual action detection tasks.

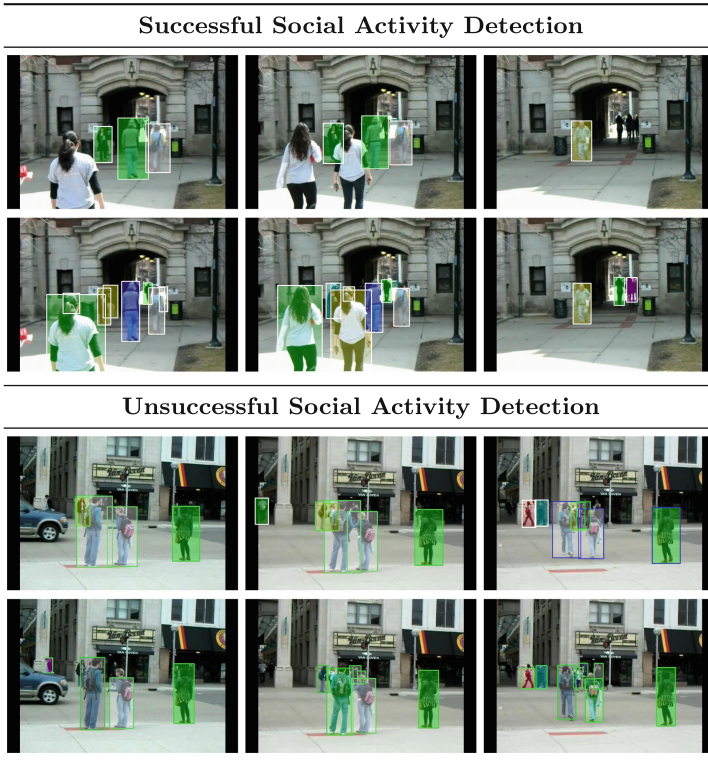
Approach	Group Activity	Indiv. Action
Ours (full model)	75.95	26.75
w/ 2 layers of temporal smoothing	72.79	23.15
w/ 2 layers of graph smoothing	73.28	22.84
w/o temporal smoothing	59.13	14.27
w/o graph smoothing	68.39	11.56
w/o action nodes	63.46	18.28
w/o $\mathcal{L}_{actor}$	69.77	12.45
w/o $\mathcal{L}_{global}$	61.28	9.28
w/ SSD instead of DETR	74.31	23.27

**Generalization to Arbitrary Action Localization** Since our approach does not make any assumptions on the number of actions or type of action, we evaluate its capability to generalize to arbitrary action localization in videos. We evaluate on the UCF Sports [36], JHMDB [16], and THUMOS’13 [17] datasets, where there is a single action in the scene with a varying number of actors. Table 3 summarizes the results, comparing our approach against other unsupervised learning baselines. We outperform the baselines on all benchmarks, except UCF Sports, when trained on videos from the same domain. The most interesting result is the top row, which shows the performance of our model, trained on CAD and evaluated on out-of-domain videos. We perform well in arbitrary action localization without explicit training, showcasing its generalization capabilities.

### 4.3 Ablation Studies

We systematically analyze the contributions of each part of our framework and quantify their effects in Table 4. We examine the presence and absence of graph smoothing, temporal smoothing, and the use of action nodes in our action graphs. We see that removing action graphs causes a dramatic decrease in group activity recognition while having minimal effect on individual action recognition. Temporal smoothing has the most impact on both metrics, which could be attributed to the fact that information from the entire video is propagated through the temporal edges and enables better contextualization of group dynamics. Graph smoothing, which allows nodes to within the same frame to share information, is

essential in propagating information from the action node to each person node. Adding additional layers of temporal and graph smoothing reduces the performance of the approach since it makes the node representations uniform and, hence, loses information about the changes in actor appearances and locations. Removing global predictions ( $\mathcal{L}_{global}$ ) results in considerably lower recognition performance (61.28%), while Removing the actor-level prediction loss ( $\mathcal{L}_{actor}$ ) results in a recognition performance of 69.77%, which is a considerable improvement over PredLearn but lower than the proposed framework. Using SSD [25] as the visual backbone, such as in PredLearn and AC-HPL, results in group activity recognition performance of 74.31% and individual action detection performance of 23.27%, lower than the proposed framework.



**Fig. 2. Qualitative visualization** successful (top) and unsuccessful (bottom) activity detection on the Collective Activities dataset. People from the same social group are highlighted in the same color, and the bounding box color indicates their social activity.

#### 4.4 Qualitative Evaluation

Figure 2 presents some qualitative visualization of the output from our framework. The top half presents successful social activity detection results. The first row is the ground truth annotations, while the second row shows our corresponding predictions. As can be seen, we can localize and recognize both the social membership (indicated by the color of the shaded region) and the social action (indicated by the bounding box color) of each actor in the scene. Interestingly, we see that we detect and recognize the social activities of people not in the ground truth (bottom left) and consistently maintain prediction throughout the sequence. The bottom half of Figure 2 shows unsuccessful results where the membership was misclassified, although the social action is correct. We attribute this to our framework’s additional action detections that provide “distractors” for the membership classification. This effect is reflected in the individual action mAP score (26.75), where the number of false alarms plays a significant role. The average recall is 67%, indicating that we can recover and label the actors.

### 5 Discussion, Limitations, and Future Work

In this paper, we presented a framework for unsupervised multi-actor, multi-action localization in streaming videos. We showed that it can be adapted to perform group activity recognition, action detection, social membership identification, and social action detection tasks in multi-actor settings. We also demonstrated its potential for localizing an arbitrary number of actions in streaming videos and showed its generalization capabilities by evaluating on out-of-domain data. While it outperformed unsupervised baselines and was competitive with supervised learning approaches, we observe some limitations that offer potential for future work. First, the actor selector module focuses on actions with unpredictable motion. Hence, it fails to consistently localize those with limited predictability, such as “waiting.” Similarly, it is sensitive to missed detections. It relies heavily on the ROI detector to provide quality region proposals. Finally, imposing constraints on group formations in frame-level action graphs will likely yield more robust social membership recognition performance. Our future work is focused on improving social action detection by dynamic graph modeling [5].

**Acknowledgements.** This work was partially supported by the US National Science Foundation Grants IIS 2348689 and IIS 2348690 and the US Department of Agriculture grant 2023-69014-39716-1030191.

### References

1. Aakur, S., Sarkar, S.: Action localization through continual predictive learning. In: ECCV. pp. 300–317 (2020)
2. Aakur, S., Sarkar, S.: Actor-centered representations for action localization in streaming videos. In: ECCV. pp. 70–87 (2022)

3. Aakur, S.N., Sarkar, S.: A perceptual prediction framework for self supervised event segmentation. In: CVPR. pp. 1197–1206 (2019)
4. Azar, S.M., Atigh, M.G., Nickabadi, A., Alahi, A.: Convolutional relational machine for group activity recognition. In: CVPR. pp. 7892–7901 (2019)
5. Bal, A.B., Mounir, R., Aakur, S., Sarkar, S., Srivastava, A.: Bayesian tracking of video graphs using joint kalman smoothing and registration. In: ECCV. pp. 440–456 (2022)
6. Bian, C., Feng, W., Wang, S.: Self-supervised representation learning for skeleton-based group activity recognition. In: ACM MM (2022)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229 (2020)
8. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
9. Choi, W., Shahid, K., Savarese, S.: What are they doing?: Collective activity classification using spatio-temporal relationship among people. In: ICCV Workshops. pp. 1282–1289 (2009)
10. Ehsanpour, M., Abedin, A., Saleh, F., Shi, J., Reid, I., Rezatofighi, H.: Joint learning of social groups, individuals action and sub-group activities in videos. In: ECCV. pp. 177–195 (2020)
11. Gavrilyuk, K., Sanford, R., Javan, M., Snoek, C.G.: Actor-transformers for group activity recognition. In: CVPR. pp. 839–848 (2020)
12. Han, M., Zhang, D.J., Wang, Y., Yan, R., Yao, L., Chang, X., Qiao, Y.: Dual-ai: Dual-path actor interaction learning for group activity recognition. In: CVPR. pp. 2990–2999 (2022)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. Hu, G., Cui, B., He, Y., Yu, S.: Progressive relation learning for group activity recognition. In: CVPR (June 2020)
15. Ibrahim, M.S., Muralidharan, S., Deng, Z., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: CVPR (June 2016)
16. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV. pp. 3192–3199 (Dec 2013)
17. Jiang, Y.G., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
18. Kim, D., Lee, J., Cho, M., Kwak, S.: Detector-free weakly supervised group activity recognition. In: CVPR. pp. 20083–20093 (2022)
19. Kim, J., Lee, M., Heo, J.P.: Self-feedback detr for temporal action detection. In: ICCV. pp. 10286–10296 (2023)
20. Kong, L., Pei, D., He, R., Huang, D., Wang, Y.: Spatio-temporal player relation modeling for tactic recognition in sports videos. *IEEE T-CSVT* **32**(9), 6086–6099 (2022)
21. Kong, L., Qin, J., Huang, D., Wang, Y., Van Gool, L.: Hierarchical attention and context modeling for group activity recognition. In: ICASSP. pp. 1328–1332 (2018)
22. Li, S., Cao, Q., Liu, L., Yang, K., Liu, S., Hou, J., Yi, S.: Groupformer: Group activity recognition with clustered spatial-temporal transformer. In: ICCV. pp. 13668–13677 (2021)
23. Li, X., Choo Chuah, M.: Sbgar: Semantics based group activity recognition. In: ICCV. pp. 2876–2885 (2017)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)





25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37 (2016)
26. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint [arXiv:1605.08104](https://arxiv.org/abs/1605.08104) (2016)
27. Lu, L., Lu, Y., Yu, R., Di, H., Zhang, L., Wang, S.: Gaim: Graph attention interaction model for collective activity recognition. *IEEE T-MM* **22**(2), 524–539 (2020)
28. Mounir, R., Vijayaraghavan, S., Sarkar, S.: Streamer: Streaming representation learning and event segmentation in a hierarchical manner. *NeurIPS* **36** (2024)
29. Pramono, R.R.A., Chen, Y.T., Fang, W.H.: Empowering relational network by self-attention augmented conditional random fields for group activity recognition. In: ECCV. pp. 71–90 (2020)
30. Qi, M., Wang, Y., Qin, J., Li, A., Luo, J., Van Gool, L.: Stagnet: An attentive semantic rnn for group activity and individual action recognition. *IEEE T-CSVT* **30**(2), 549–565 (2019)
31. Qi, M., Wang, Y., Qin, J., Li, A., Luo, J., Van Gool, L.: stagnet: An attentive semantic rnn for group activity and individual action recognition. *IEEE T-CSVT* **30**(2), 549–565 (2020)
32. Raviteja Chappa, N.V., Nguyen, P., Nelson, A.H., Seo, H.S., Li, X., Dobbs, P.D., Luu, K.: Sogar: Self-supervised spatiotemporal attention-based social group activity recognition. arXiv e-prints pp. arXiv–2305 (2023)
33. Shu, T., Todorovic, S., Zhu, S.C.: Cern: confidence-energy recurrent network for group activity recognition. In: CVPR. pp. 5523–5531 (2017)
34. Shu, X., Zhang, L., Sun, Y., Tang, J.: Host–parasite: Graph lstm-in-lstm for group activity recognition. *IEEE TNNLS* (2021)
35. Soomro, K., Shah, M.: Unsupervised action discovery and localization in videos. In: ICCV. pp. 696–705 (2017)
36. Soomro, K., Zamir, A.R.: Action recognition in realistic sports videos. In: *Computer Vision in Sports*, pp. 181–208. Springer (2015)
37. Tamura, M., Vishwakarma, R., Vennelakanti, R.: Hunting group clues with transformers for social group activity recognition. In: ECCV. pp. 19–35 (2022)
38. Tang, J., Shu, X., Yan, R., Zhang, L.: Coherence constrained graph lstm for group activity recognition. *IEEE T-PAMI* **44**(2), 636–647 (2019)
39. Tang, Y., Lu, J., Wang, Z., Yang, M., Zhou, J.: Learning semantics-preserving attention and contextual interaction for group activity recognition. *IEEE T-IP* **28**(10), 4997–5012 (2019)
40. Tarashima, S.: One-shot deep model for end-to-end multi-person activity recognition. In: *BMVC* (2021)
41. Thilakarathne, H., Nibali, A., He, Z., Morgan, S.: Group activity recognition using unreliable tracked pose. arXiv preprint [arXiv:2401.03262](https://arxiv.org/abs/2401.03262) (2024)
42. Trehan, S., Aakur, S.N.: Towards active vision for action localization with reactive control and predictive learning. In: WACV. pp. 783–792 (2022)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *NeurIPS* **30** (2017)
44. Wang, M., Ni, B., Yang, X.: Recurrent modeling of interaction context for collective activity recognition. In: CVPR. pp. 3048–3056 (2017)
45. Wu, J., Wang, L., Wang, L., Guo, J., Wu, G.: Learning actor relation graphs for group activity recognition. In: CVPR. pp. 9964–9974 (2019)
46. Wu, L., Lang, X., Xiang, Y., Chen, C., Li, Z., Wang, Z.: Active spatial positions based hierarchical relation inference for group activity recognition. *IEEE T-CSVT* (2022)

47. Xie, Z., Jiao, C., Wu, K., Guo, D., Hong, R.: Active factor graph network for group activity recognition. *IEEE T-IP* (2024)
48. Yan, R., Xie, L., Tang, J., Shu, X., Tian, Q.: Higcin: Hierarchical graph-based cross inference network for group activity recognition. *IEEE T-PAMI* **45**(6), 6955–6968 (2020)
49. Yan, R., Xie, L., Tang, J., Shu, X., Tian, Q.: Social adaptive module for weakly-supervised group activity recognition. In: *ECCV*. pp. 208–224 (2020)
50. Yuan, H., Ni, D.: Learning visual context for group activity recognition. In: *AAAI Conference on Artificial Intelligence*. vol. 35, pp. 3261–3269 (2021)
51. Zacks, J.M., Tversky, B., Iyer, G.: Perceiving, remembering, and communicating structure in events. *J. Exp. Psychol. Gen.* **130**(1), 29 (2001)
52. Zhang, P., Tang, Y., Hu, J.F., Zheng, W.S.: Fast collective activity recognition under weak supervision. *IEEE T-IP* **29**, 29–43 (2019)
53. Zhou, H., Kadav, A., Shamsian, A., Geng, S., Lai, F., Zhao, L., Liu, T., Kapadia, M., Graf, H.P.: Composer: compositional reasoning of group activity in videos with keypoint-only modality. In: *ECCV*. pp. 249–266 (2022)



# Joint-Temporal Action Segmentation via Multi-action Recognition

Usfita Kiftiyani  and Seungkyu Lee <sup>(✉)</sup> 

Kyung Hee University, Seoul, South Korea  
{kiftiyani, seungkyu}@khu.ac.kr

**Abstract.** Action recognition and segmentation are critical tasks for the applications requiring detailed analysis on human behavioral characteristics. However, current research primarily concentrates on temporal action segmentation assuming sequential occurrences of sub-actions. In practice, multiple actions temporarily overlapped or even co-occur in parallel. Inspired by image segmentation methods, we propose a joint-temporal action segmentation method that performs multi-action recognition at each human body joint. To conduct quantitative and qualitative evaluations, we construct a new skeleton-based multi-action dataset from the existing N-UCLA dataset (The code for our data generation is available at <https://github.com/kiftiyani/NUCLAOverlap.git>). We propose learning objectives that incorporate the class distribution of each point to address the continuous label problem. Additionally, we argue that the inter-dependency between joints is crucial. We conduct multi-action segmentation experiments comparing well-known objectives such as CE, MAE, and MSE. Evaluation results demonstrate that our proposed approach achieves outstanding performance on five backbones.

**Keywords:** Fine-grained action · Multi-action decomposition · Action regression · Joint dependency · Mutual information

## 1 Introduction

Human action recognition and segmentation are essential tasks in various computer vision applications, including but not limited to healthcare services, sports, surveillance, and human-computer interaction. For instance, the quality of camera-guided healthcare services is critically influenced by the performance of human body motion understanding and behavioral action monitoring. In regular human actions, multiple fine-grained small actions compose a longer, coarse-grained action [23]. For example, engaging in a basketball game encompasses small actions such as running, dribbling, jumping, and throwing a ball. In this case, playing basketball is a type of broader human action while the involved small actions collectively describe the characteristics of the broad action.

Skeleton-based human action recognition [5, 9, 13, 16, 25, 26, 28, 34] has been widely studied thanks to the development of depth sensors and body tracking

algorithms. Earlier approaches either represent the skeleton as a sequence of joint coordinates and model its high-level features [28] in an unsupervised manner with recurrent neural networks (RNNs) or use a hand-crafted representation [16] of the human body as the input to convolutional neural networks (CNNs)-based prediction methods. However both approaches do not fully recognize the human body structure in the skeleton data. Yan et al. [34] propose to automatically capture the patterns embedded in the spatial configuration of the human joints using graph representation and expand it to the temporal dynamics in collective manner. This achieves substantial improvements over the conventional methods and has triggered many graph based approaches to better capture the topology of actions [5, 9, 13, 22, 25, 26].

Most existing approaches recognize an entire body sequence as a single action class. However, in real-world applications, an action may involve only a subset of joints in each temporal frame rather than the entire body of the whole sequence. Moreover, multiple actions can occur simultaneously at different joint parts. This highlights the importance of developing more fine-grained action recognition methods that can accurately identify actions based on specific joint movements or combinations of joints. Research in fine-grained action recognition [1, 19, 21, 35] predominantly focuses on video-based temporal action segmentation, which involves dissecting sequences into multiple sub-actions temporally. Therefore, achieving fine-grained action segmentation at the joint-temporal level poses a greater challenge compared to traditional temporal action segmentation.

Inspired by semantic image segmentation, we aim to classify actions at each individual joint-temporal position. The main goal, akin to general image segmentation, is to categorize the action type of each joint based on its semantic meaning [7]. Research on transformer-based image segmentation [6, 15] commonly employs the widely used cross-entropy (CE) loss which measures pixel-wise errors independently as their learning objective, resulting in commendable accuracy. A lot of research on image segmentation note that segmentation errors tend to occur predominantly near the boundaries. This has brought considerable interest in recognizing the significance of boundary information for enhancing segmentation performance [4, 29, 32]. On the other hand, segmentation boundary between human actions in joint-temporal space does not have to be clear because human body joints are physically connected. Two neighboring joints cannot show two separate and distinct movements. This indicates that the action category of a joint includes multiple actions with different weights rather than a single action type.

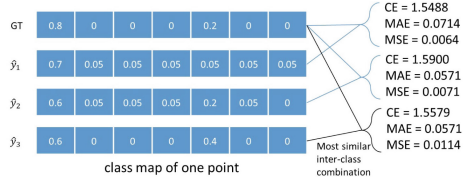
Unfortunately, there is no available benchmark dataset for joint-temporal action segmentation due to the difficulty of manually labeling each joint action type. To resolve this issue and achieve an appropriate dataset for quantitative evaluation, we propose to build a new dataset for multi-action recognition and segmentation in the joint-temporal domain from existing skeleton-based datasets. To this end, we estimate joint-wise importance for the corresponding action type of the samples and combine two samples of different actions with respective importance weights. Weighted labels of multiple actions assigned to

each joint indicate that a joint is involved in more than one action, performing a series of smoothly connected activities. Finally, our data labels are continuous values indicating the amount of contribution to each action type.

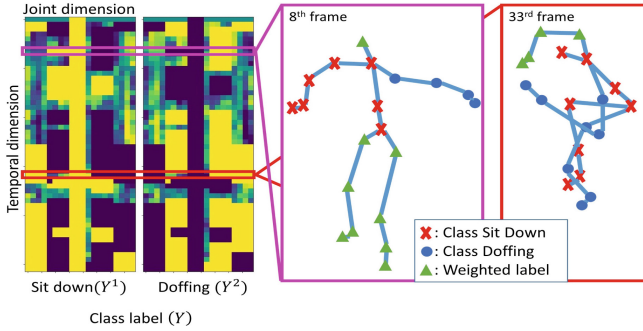
To perform multi-action segmentation, we investigate the performance of well-known learning objectives, such as the cross-entropy (CE) loss, in tackling our specific task. Additionally, considering the similarity of our task to a regression problem, we investigate the effectiveness of mean absolute error (MAE) and mean squared error (MSE). The conventional learning objectives (CE, MAE, MSE loss) calculate the discrete value of each class in every point between the estimated and ground truth labels without considering the class distribution of each point. As shown in Fig. 1, these objectives cannot effectively measure the class distribution of a point. Our approach focuses on incorporating the class distribution of each joint-temporal-wise point.

Specifically, we design an objective function, namely **independent point class distribution (IPCD) loss**, that consider the class distribution by measuring the vector similarity between points in the ground truth and their corresponding points in the estimated label. Additionally, based on our observation as depicted in the left part of Fig. 2, there exists an inter-dependency between joints. Some joints demonstrate consistent class patterns, while others exhibit significantly different class patterns across the temporal dimension. This reminds us how joint movement can be influenced by the neighboring joints of the same body part. Building on this insight, we propose a new objective function called **joint dependency loss** to incorporate joint dependency information, which improve the performance of the independent point objective, especially in the context of multiple actions.

We conduct experimental evaluations on skeleton-based multi-action segmentation and compare our results with well-known learning objectives (CE, MAE and MSE loss) on our dataset using five popular backbones in skeleton-based action recognition: attention-enhanced adaptive graph convolutional neural network (AAGCN) [25, 26], channel-wise topology refinement graph convolutional network (CTRGCN) [5], information bottleneck-based graph convolutional network (InfoGCN) [9], Multi-scale spatial-temporal convolutional neural network (MSSTNet) [8], and temporal decoupling graph convolutional network (TD\_GCN) [22]. The experimental results show that our proposed learning objectives achieve state-of-the-art performance in terms of root mean square error (RMSE). Our analysis reveals that relying solely on an independent point-wise learning objective is insufficient for the model to learn continuous class labels effectively. However, incorporating joint dependency information notably enhances performance, particularly in multi-action cases.



**Fig. 1.** Conventional loss measurement between two points with different class distribution (smaller better).



**Fig. 2.** Multi-action label example from our proposed dataset for merged action class sit down and doffing. Left figures is the class label visualization of the existing classes (dark to bright colors indicate label value between 0 (darkest color) and 1 (brightest color)). Right figures are examples of skeleton body structure in selected frames with class label illustration in every joints.

The main contributions of our work are as follows: (1) we propose a basic learning objective, IPCD loss, which handles continuous labels by incorporating the class distribution of each point independently. (2) We propose additional learning objective based on the inter-joint dependency information namely joint dependency loss, which enhances the model’s ability to classify the multi-action region. (3) To conduct the desired experiment, we develop a new multi-action label skeleton-based human action dataset (merged N-UCLA) from the existing Northwestern-UCLA (N-UCLA) [30] dataset.

## 2 Multi-action Dataset

The previous work [23] defines coarse- and fine-grained actions by identifying that multiple fine-grained actions in one sequence build a single coarse-grained action representing the same context. Here, we expand this definition by considering the nature of real actions, where a sequence of actions can be composed of multiple fine-grained actions representing different contexts. For example, someone can walk around while carrying something. The actions of walking and carrying something are two independent actions that happen simultaneously but do not necessarily build the same context. With this expanded definition, we argue that decomposing actions for each body joint would be advantageous for future research in fine-grained behavioral action analysis, as it provides detailed detection and localization of specific actions. While existing fine-grained action segmentation datasets, such as Breakfast [18], 50Salads [27], and JIGSAW [10], focus on specific parts of the human body that reveal core movements of each action class, observing behavioural actions from the entire human body can offer distinct advantages.

Thus, we utilize skeleton-based human action datasets that provide a full human body structure due to its nature of free from noisy background. Specif-

ically, we utilize the N-UCLA [30] dataset, that provides 3D human skeleton data captured by a Kinect camera. The dataset comprises 1494 samples of 10 action classes listed in Table 1a. From the existing class in N-UCLA, we pair combinations of two actions and select the most realistic combination based on our judgement and the appearance of the data. For example, since the "walk around" class predominantly involves the lower part of the body, it would be realistic to merge it with an action primarily involving the upper body. Due to the limitation of the class labels and the importance of realistic combinations, this study only merges pairs of two classes listed in Table 1b.

**Table 1.** a) Regular N-UCLA action labels. b) Possible realistic action combination of N-UCLA for our proposed dataset.

Class Index	Action Name	Class Composition	Action Name
1	Pick up with one hand	4, 3	Walk around while drop trash
2	Pick up with two hand	4, 9	Walk around while throw
3	Drop trash	4, 10	Walk around while carry
4	Walk around	5, 7	Sit down while donning
5	Sit down	5, 8	Sit down while doffing
6	Stand up	5, 10	Sit down while carry
7	Donning	6, 7	Stand up while donning
8	Doffing	6, 8	Stand up while doffing
9	Throw	6, 9	Stand up while throw
10	Carry	6, 10	Stand up while carry

Initially, to guide the joint-wise labeling, we partition the body into six parts based on proximity and the nature of body movement coordination: head, spine, left arm, right arm, left leg, and right leg. We calculate the average motion of each body part to determine the class label for each joint within these body parts. Let  $X \in \mathbb{R}^{T \times J \times 3}$  define an instance of action sequence, where  $T$  and  $J$  are the temporal and joint dimension, respectively. Then  $M \in \mathbb{R}^{T \times J}$  define the motion value of  $X$ . The label value of existing class  $Y^c$  defined by:

$$Y_{j,t}^1 = \begin{cases} 1, & |\bar{m}(1)_{p,t} - \bar{m}(2)_{p,t}| \geq \tau \text{ and } \bar{m}(1)_{p,t} > \bar{m}(2)_{p,t} \\ \frac{m(1)_{j,t}}{m(1)_{j,t} + m(2)_{j,t}}, & |\bar{m}(1)_{p,t} - \bar{m}(2)_{p,t}| < \tau \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

We illustrate the generated labels  $Y^1$  for "Sit down" and  $Y^2$  for "Doffing" in Fig. 2, with the label values of  $Y^1$  measured by Eq. 1.  $Y_{j,t}^c$  indicates the label value for class  $c$  at joint  $j$  and frame  $t$ , and  $m(c)_{j,t}$  indicates motion value of an instance with class label  $c$  at joint  $j$  and frame  $t$ . Additionally,  $|\cdot|$  denoting the absolute value operation. The  $\bar{m}(c)_{p,t}$  indicates the mean of all joints' motion corresponding to body part  $p$  and frame  $t$  for instance with class  $c$ . We set

$\tau = 0.01$  to determine the weighted label regions. Then, the values of label  $Y^2$  obtained by  $Y^2 = 1 - Y^1$ .

In the example in Fig. 2, considering the combination of actions "Sit down" and "Doffing", the red X in the 8<sup>th</sup> and 33<sup>rd</sup> frames indicates joint as label "Sit down", with label value of 1 (brightest color in the "Sit down" class label) and 0 (darkest color in class label "Doffing"). Conversely the blue circle indicates joints labeled as "Doffing". Additionally, some joints may have two existing labels, denoted by the green triangle in Fig. 2, with weighted values calculated using Eq. 1 for each label. The weight values represented by colors ranging from dark to bright yellow in Fig. 2 indicating values between 0 and 1.

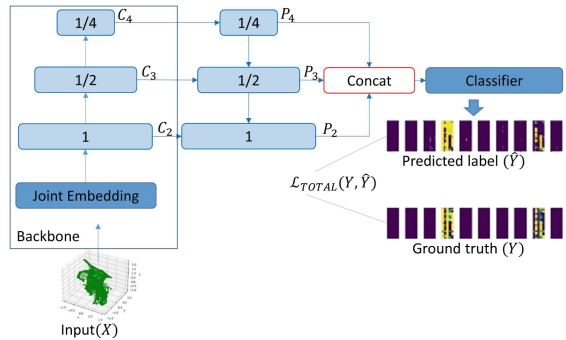
We sample a maximum of 200 samples for each combination, resulting in a total of 1983 samples.

### 3 Feature Pyramid Network for Multi-action Recognition

We illustrate our network in Fig. 3 based on Feature Pyramid Network (FPN) [20] and following the segmentation head of UPerNet (Unified Perceptual Parsing Network) [33] for simplicity. FPN serves as a versatile feature extractor that leverages the inherent multi-scale and pyramidal hierarchy within deep convolutional networks. It employs a top-down architecture with lateral connections to construct high-level semantic feature maps across all scales. While UPerNet extends the FPN architecture into a multi-task framework capable of effectively segmenting a broad range of concepts in scene understanding, including dense object detection.

We adopt FPN architecture due to its adaptability to multi-scale feature hierarchy present in most CNNs backbones. We select backbones that have similar structure to Residual Networks [11], which utilize multiple layers with the same feature map size at each stage. Here, we denote the last feature maps of each stage in the backbone as  $\{C_2, C_3, C_4\}$ , and the output feature maps of each layer in the top-down FPN as  $\{P_2, P_3, P_4\}$ . The down sampling rates are  $\{1, 2, 4\}$ , respectively.

Finally, we follow UPerNet [33] by fusing all feature



**Fig. 3.** We evaluate our proposed objective functions on a simple network structure based on Feature Pyramid Network (FPN) [20] and UPerNet [33]. Each stage in the bottom-up FPN follows the backbone's hierarchical structure, with feature map sizes denoted by  $C_2, C_3, C_4$  at scale 1, 1/2, 1/4, respectively. Each stage in the top-down FPN and the classification head follow the structure of UPerNet [33]. The objective functions measured from the predicted label and the ground truth label with  $\mathcal{L}_{TOTAL}(Y, \hat{Y})$  defined in Section 4.



maps from FPN. These feature maps are resized via bilinear interpolation to match the size of  $P_2$ , and concatenated. Subsequently, a  $1 \times 1$  convolution layer is applied to fuse features from different levels as well as reduce the channel dimensions. We introduce this straightforward network to emphasize our focus on analyzing multi-action recognition learning objective.

## 4 Methods

Existing conventional dense prediction learning objectives typically calculate loss based on either the discrete values of estimated classification maps (e.g., CE, MAE, and MSE) or estimated class regions (e.g., mIoU (mean intersection over union) and Dice loss). However, given the characteristics of our dataset—particularly its continuous labels and the inter-dependencies between joints—we argue that region-based learning objectives are unsuitable. Specifically, our dataset includes continuous labels that represent ambiguous or mixed actions, making it challenging to apply conventional loss functions effectively.

**Independent Point Class Distribution (IPCD).** Appropriate class distribution information can significantly enhance the learning process. Existing learning objectives typically penalize independent class distribution of each point by focusing on discrete class labels. Unlike segmentation tasks where pixels are often clustered based on feature similarity using contrastive learning [14, 31], our approach focuses on joint-temporal action segmentation where points may belong to multiple classes without clear boundaries between them. Thus, we introduce continuous class labels with Gaussian distribution in our dataset. To address this issue, we propose a simple objective function that considers the global class distribution of points rather than independent discrete values of each class indication. This function aims to calculate the correlation between corresponding points in the estimated label and the ground truth.

We denote an estimated class map and its corresponding ground truth as  $\hat{Y}, Y \in \mathbb{R}^{T \times J \times N_{class}}$ , respectively <sup>1</sup>. Then the independent point class distribution loss,  $\mathcal{L}_{IPCD}$ , is defined by:

$$\mathcal{L}_{IPCD} = \frac{1}{TJ} \sum_{i \in T, J} \exp |(y_i + \epsilon) \cdot (y_i + \epsilon) - (\hat{y}_i + \epsilon) \cdot (\hat{y}_i + \epsilon)| \quad (2)$$

In Eq. 2,  $\epsilon$  is a small value used to prevent *non-classes* <sup>2</sup> regions from being ignored. The  $\epsilon$  also support our introduced class distribution with assuming that each zero value in the label as units with really small value. The expression  $(\hat{y}_i + \epsilon) \cdot (y_i + \epsilon)$  computes the dot product similarity between each point in the estimated label and the corresponding point in the ground truth based on the

<sup>1</sup>  $T$ ,  $J$ , and  $N_{class}$  denote the data temporal dimension, joint dimension and class number, respectively.

<sup>2</sup> We use term *non-class* to indicate the label with zero value in the mask and term *class* to indicate non-zero labels.

class distribution, where  $y_i, \hat{y}_i \in \mathbb{R}^{N_{class}}$  and  $i$  denotes an index in joint-temporal dimension. In geometric terms, it is important to note that the maximum value of the dot product does not necessarily indicate that both vectors are the most similar. Instead, it measures the alignment of vectors with higher magnitudes. Therefore, solely maximizing the dot product between the estimated and ground truth labels is not sufficient for penalizing our model effectively. Instead, we interpret the dot product measurement as the degree of correlation between the two vectors. To achieve our objective, we also evaluate the self-correlation of each point in the ground truth  $(y_i + \epsilon) \cdot (y_i + \epsilon)$  and aim to align the estimated-to-ground truth correlation with the ground truth self-correlation degree. Thus, our independent point class distribution loss,  $\mathcal{L}_{IPCD}$ , works to align the estimated class distribution pattern with the ground truth class distribution pattern for each point. This alignment helps the model to predict low distribution in *non-class* regions while improving the *class* distribution overall.

**Joint Dependency.** Theoretically, optimizing each point independently should yield good results in estimating the multi-action continuous label. However, in practice, this approach fails to effectively detect the multi-action region. To address this issue and leverage the characteristics of our labels, we propose to incorporate the joint dependency information. As shown in the action labels in Fig. 2, there is a certain degree of similarity in class distribution between joints along the temporal dimension, which we refer to as joint dependency. By considering all class information across the temporal dimension for each joint, the joint dependency measurement aims to capture the coherence of multi-action regions. Therefore, joint dependency is expected to enhance the estimation, particularly in regions where multiple actions overlap. We propose to quantify the degree of joint dependency inspired by Mutual Information (MI) calculation. This metric will help us assess how information shared between joints can improve the accuracy of multi-action recognition.

Mutual information is closely related to the statistical dependency between variables [17]. Exact computation of mutual information is typically feasible only for discrete variables (where the sum can be computed exactly) or when probability distributions are known [3]. Previous work has often optimize learning representation by maximizing the mutual information between strongly correlated components, such as input-output [12] or across view [2], where computing distributions directly is challenging. In our approach, we aim to leverage the principles of mutual information calculation to learn the dependency information between two known distributions, specifically the estimated labels between joints. We calculate the mutual information between two joints  $A \in \mathbb{R}^T$  and  $B \in \mathbb{R}^T$  by the following formulation:

$$MI(A; B) = \frac{1}{T} \sum_{t \in T} a_t \cdot b_t \log \frac{a_t \cdot b_t}{\mu(A)\mu(B)} \quad (3)$$

note that  $a_t$  and  $b_t$  are vectors with class labels. From the formulation, mutual information is defined by weighting the correlation between two different joints

from the same temporal index  $a_t \cdot b_t$  by the correlation information over the global class distribution of each joint  $\frac{a_t \cdot b_t}{\mu(A)\mu(B)}$ . Thus, our proposed joint dependency loss  $\mathcal{L}_{dep}$  is defined by:

$$\mathcal{L}_{dep} = |Y_{MI} - \hat{Y}_{MI}| \quad (4)$$

where  $Y_{MI}$  and  $\hat{Y}_{MI}$  denote the mutual information between joints in  $Y$  and  $\hat{Y}$ , respectively.

Finally, our total loss is calculated as follows:

$$\mathcal{L}_{TOTAL} = \mathcal{L}_{IPCD} + \lambda \mathcal{L}_{dep} \quad (5)$$

where  $\lambda$  represents the weight for the joint dependency loss  $\mathcal{L}_{dep}$ .

## 5 Experimental Evaluation

We compare our results with conventional objective functions and conduct ablation studies to examine the effect of each objective quantitatively. Additionally, we provided qualitative insights by visually inspecting the decomposition outcomes. This involved showcasing the predicted labels and the decomposed actions to illustrate the efficacy of our approach.

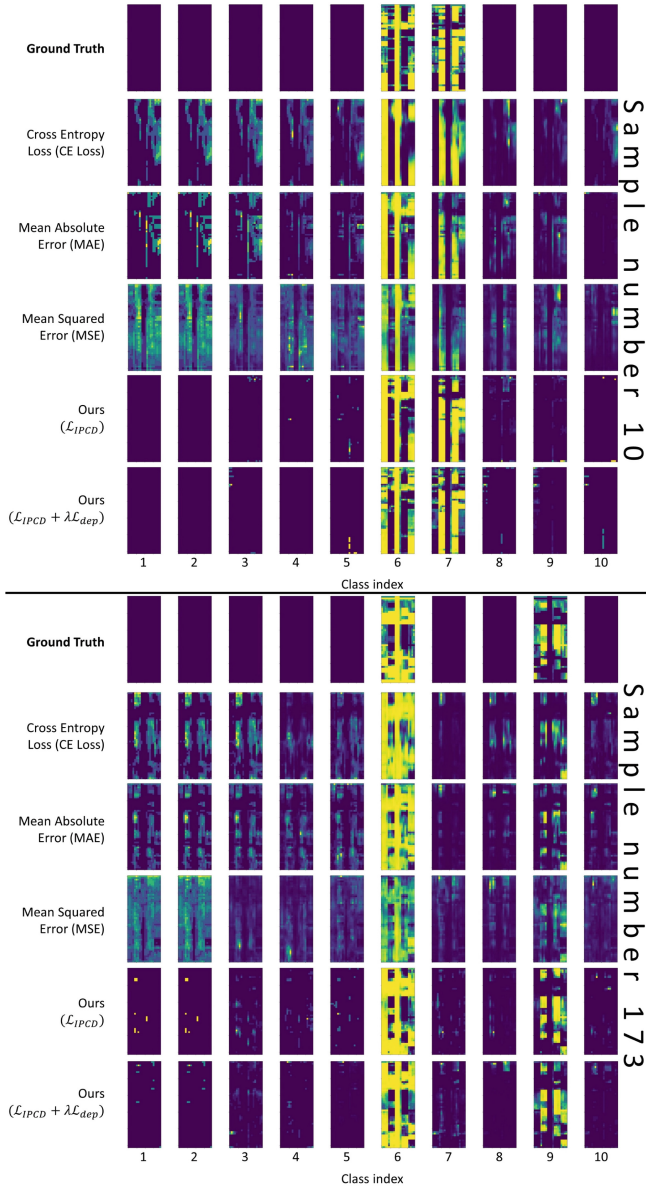
### 5.1 Dataset

The merged N-UCLA dataset is our newly created dataset by merging two samples from different classes of the human skeleton dataset N-UCLA [30]. The merging process is explained in detail in Section 2. This dataset composed of 10 action classes and contains 1983 samples in total. We divide the data for training and testing by 80:20 ratio.

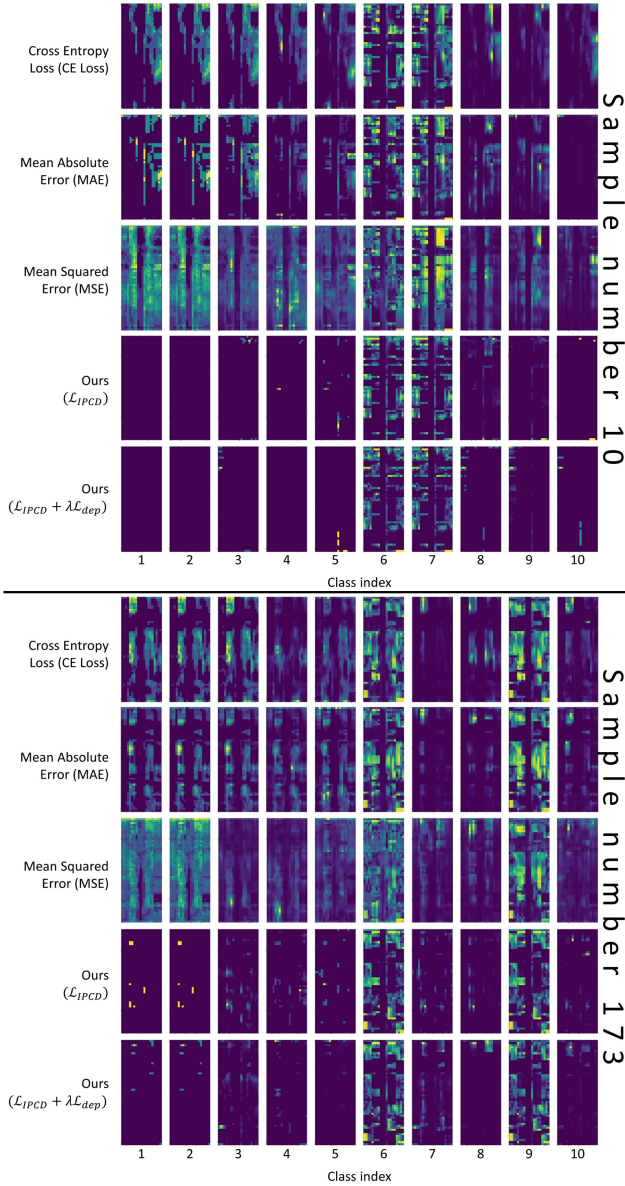
### 5.2 Implementation Details

**Backbone.** We employ five state-of-the-art models in skeleton-based human action recognition as our backbone: AAGCN [26], CTRGCN [5], InfoGCN [9], TD\_GCN [22], and MSSTNet [8]. The first four models are GCN-based, specializing in capturing the semantic context of human action by leveraging the topology of human body movements. MSSTNet [8], on the other hand, is the most recent CNN-based model in skeleton-based action recognition. For simplicity, we employ the single-stream version of these backbones, utilizing only joint information as the input.

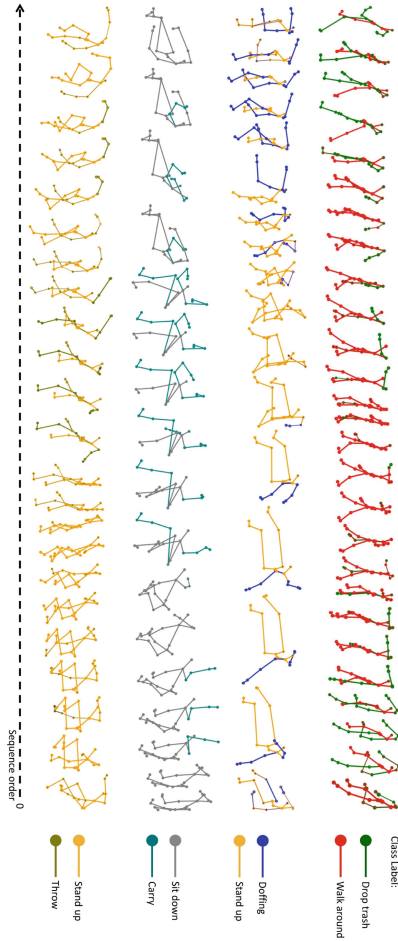
**Training Setting.** We conduct all experiments with 110 epochs and implementing a warm-up strategy for the first 5 epochs following the method described in [9]. We set the weight decay to  $4 \times 10^{-4}$  and use a batch size of 16. Based on the empirical experiments, we define different weight  $\lambda$  for each backbone in Eq. 5. All backbones were pretrained on the NTU-RGB+D 60 [24] dataset, where we augment the joint structure from 25 to 20 joints to align with the human skeleton structure of the N-UCLA dataset.



**Fig. 4.** Qualitative comparison of the result prediction label values between conventional and our proposed loss function with CTRGCN [5] backbone. In this figure, we display all 10 class indices to demonstrate the performance of each objective function in multi-action prediction, specifically across both *class* and *non-class* regions. Note that the labels are represented with continuous values, ranging from the darkest to the brightest colors, indicating values from 0 to 1, respectively.



**Fig. 5.** The absolute different between the ground truth and prediction label values from Fig. 4 samples. This value indicates the prediction error for each class indication in the joint-temporal dimension. This figure demonstrates that our proposed objective functions predict the *non-class* regions with smaller error (dark color) and exhibit similar error distribution compared to the baseline methods.



**Fig. 6.** Examples of action decomposition results (best viewed in color). Based on our experimental assumptions, each sample involves a maximum of two actions. We select the top-2 predicted labels to visualize the decomposition results for each joint. The thickness of the points and edges corresponds to the weight of the labels, with thicker lines or points indicating larger weights. For example, the skeleton sequence in the first row shows that the "drop trash" action is dominated by hand movements, represented by green-colored joints and edges. In contrast, the "walk around" action is characterized by movements of the spine and lower body parts, represented by red-colored joints and edges. The figure also illustrates mixed actions, where some joints or edges are indicated by two colors of different thicknesses, representing the weighted label value. In the second row, the sequence combines the actions "doffing" and "stand up". It demonstrates that "doffing" action is performed predominantly by hand movements, while "stand up" action involves movements of the spine and lower body parts. And mixed labels are visible in some hand parts, indicating their involvement in both "doffing" and "stand up" actions. Similarly, in rows three and four, we can observe which body parts are involved in each action.

### 5.3 Experimental Results

We compare our results on five backbones: AAGCN [25,26], CTRGCN [5], InfoGCN [9], TD\_GCN [22], and MSSTNet [8]; with existing conventional objective functions in Table 2 by calculating the root mean squared error (RMSE). Across all five models, our proposed objective functions achieves state-of-the-art performance, validating the effectiveness of our work.

**Table 2.** Comparative results on the Merged N-UCLA dataset. All results are obtained with the same training settings, except for the weight parameter  $\lambda$  in Eq. 5 for each backbone: 0.7 for AAGCN, 0.6 for CTRGCN, 0.4 for InfoGCN, and 0.5 for both MSSTNet and TD\_GCN.

Objective Func.	RMSE				
	AAGCN [26]	CTRGCN [5]	InfoGCN [9]	TD_GCN [22]	MSSTNet [8]
CE	0.1332	0.1268	0.1317	0.1218	0.1306
MAE	0.1145	0.1192	0.1269	0.1169	0.1154
MSE	0.1143	0.1100	0.1102	0.0983	0.1147
Ours (w/o $\mathcal{L}_{dep}$ )	0.1124	0.1006	0.1122	0.093	0.1085
Ours ( $\mathcal{L}_{TOTAL}$ )	<b>0.1066</b>	<b>0.095</b>	<b>0.1052</b>	<b>0.0796</b>	<b>0.1046</b>

**Table 3.** RMSE comparison of *class* and *non-class* regions. This comparison shows the model’s performance in multi-label prediction. By analyzing the error in label values between *class* and *non-class* regions, we demonstrate that our proposed  $\mathcal{L}_{IPCD}$  loss improves the predictions significantly in the *class* regions compared to the baseline methods. Furthermore, incorporating the proposed  $\mathcal{L}_{dep}$  further enhances predictions in both *class* and *non-class* regions.

Objective Func.	RMSE ( <i>class</i> $\pm$ <i>non-class</i> )				
	AAGCN [26]	CTRGCN [5]	InfoGCN [9]	TD_GCN [22]	MSSTNet [8]
CE	0.3543 $\pm$ 0.0763	0.3410 $\pm$ 0.0707	0.3486 $\pm$ 0.0764	0.3279 $\pm$ 0.0677	0.3457 $\pm$ 0.0756
MAE	0.2926 $\pm$ 0.0714	0.3178 $\pm$ 0.0680	0.3310 $\pm$ 0.0760	0.3119 $\pm$ 0.0664	0.2937 $\pm$ 0.0725
MSE	0.3074 $\pm$ <b>0.0637</b>	0.2966 $\pm$ 0.0608	0.2944 $\pm$ <b>0.0624</b>	0.2666 $\pm$ 0.0536	0.308 $\pm$ <b>0.0642</b>
Ours (w/o $\mathcal{L}_{dep}$ )	0.2852 $\pm$ 0.0711	0.2587 $\pm$ 0.0621	0.2833 $\pm$ 0.0716	0.2441 $\pm$ 0.0548	0.2759 $\pm$ 0.0683
Ours ( $\mathcal{L}_{TOTAL}$ )	<b>0.2698</b> $\pm$ 0.0679	<b>0.2422</b> $\pm$ <b>0.0595</b>	<b>0.2673</b> $\pm$ 0.0663	<b>0.2092</b> $\pm$ <b>0.0469</b>	<b>0.2614</b> $\pm$ 0.0679

**Independent point class distribution (IPCD) loss.** We demonstrate the impact of utilizing our IPCD loss from Eq. 2. Table 2 compares our method to the baseline methods. Compared to the best baseline method, MSE, our method shows improvements in most of the models except for InfoGCN. Significant improvements are also observed in the *class* regions, as detailed in Table 3.

Although the quantitative evaluation of the *non-class* regions shown in Table 3 does not exhibit meaningful improvement compared to MSE, visualization in Fig. 4 and Fig. 5 indicate that compared to the baseline methods our IPCD loss predict the *non-class* regions in near zero value almost evenly. This suggests that our proposed IPCD loss provides an alternative solution to conventional loss functions in addressing weighted label problems. However, it remains insufficient for decomposing the multi-action region effectively.

**Joint dependency loss.** As described in Section 4, we introduce the joint dependency loss to complement the class distribution loss. By incorporating dependencies between joints, the model is encouraged to learn temporal-wise class patterns, thereby aiding in the recognition of multi-action labels. The last row of Table 2 shows that our method outperforms the baselines across all backbones. Furthermore, significant improvements are observed in both *class* and *non-class* regions than without joint dependency loss, as shown in the last two rows of Table 3. This is further supported by the action label prediction in the last row of Fig. 4, where our method effectively portrays the multi-action regions better than other methods. This supports our hypothesis that leveraging joint dependency information alongside class distribution information helps the models to effectively learn multi-action regions with minimal errors, as demonstrate in Fig. 5, where errors are measured by the absolute different between the ground truth and the predicted labels. Finally, our decomposition results are illustrated in Fig. 6, showing how each action is decomposed in joint-temporal-wise manner based on the action recognition results.

## 6 Conclusion

This study demonstrates the feasibility of action segmentation in joint-temporal domain. We introduce novel learning objectives, namely independent point class distribution (IPCD) loss and joint dependency loss, which have been shown to effectively learn continuous action labels. Remarkably, our method achieves state-of-the-art performance compared to existing conventional loss functions on merged N-UCLA dataset using a simple network architecture.

**Acknowledgements.** This work was supported by the National Research Foundation of Korea (NRF) grant (No.NRF-2020R1A2C1015146), the IITP (Institute of Information & Communications Technology Planning&Evaluation) grant (Artificial Intelligence Innovation Hub, RS-2021-II212068) funded by the Korea government (Ministry of Science and ICT), and the Korean Council for University Education (KCUE).

## References

1. van Amsterdam, B., Kadkhodamohammadi, A., Luengo, I., Stoyanov, D.: Aspnet: Action segmentation with shared-private representation of multiple data sources. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2384–2393 (June 2023)



2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA (2019)
3. Belghazi, M.I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, D.: Mutual information neural estimation. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 531–540. PMLR (10–15 Jul 2018)
4. Borse, S., Wang, Y., Zhang, Y., Porikli, F.: Inverseform: A loss function for structured boundary-aware segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5901–5911 (June 2021)
5. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13359–13368 (October 2021)
6. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. In: The Eleventh International Conference on Learning Representations (2023)
7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1290–1299 (June 2022)
8. Cheng, Q., Cheng, J., Ren, Z., Zhang, Q., Liu, J.: Multi-scale spatial-temporal convolutional neural network for skeleton-based action recognition. *Pattern Anal. Appl.* **26**, 1303–1315 (2023)
9. Chi, H.g., Ha, M.H., Chi, S., Lee, S.W., Huang, Q., Ramani, K.: Infocgn: Representation learning for human skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20186–20196 (June 2022)
10. Gao, Y., Vedula, S.S., Reiley, C.E., Ahmidi, N., Varadarajan, B., Lin, H.C., Tao, L., Zappella, L., Béjar, B., Yuh, D.D., et al.: The jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In: MICCAI workshop: M2cai. vol. 3 (2014)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
12. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: International Conference on Learning Representations (2019)
13. Huang, X., Zhou, H., Wang, J., Feng, H., Han, J., Ding, E., Wang, J., Wang, X., Liu, W., Feng, B.: Graph contrastive learning for skeleton-based action recognition. In: The Eleventh International Conference on Learning Representations (2023)
14. Huang, Y., Kang, D., Chen, L., Zhe, X., Jia, W., Bao, L., He, X.: Car: Class-aware regularization for semantic segmentation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision - ECCV 2022*, pp. 518–534. Springer Nature Switzerland, Cham (2022)
15. Jain, J., Singh, A., Orlov, N., Huang, Z., Li, J., Walton, S., Shi, H.: Semask: Semantically masked transformers for semantic segmentation. In: *ICCV Workshops 2023* (2023)

16. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4570–4579 (2017)
17. Kinney, J.B., Atwal, G.S.: Equitability, mutual information, and the maximal information coefficient. *Proc. Natl. Acad. Sci.* **111**(9), 3354–3359 (2014)
18. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 780–787 (2014)
19. Lea, C., Reiter, A., Vidal, R., Hager, G.D.: Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9907, pp. 36–52. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46487-9\\_3](https://doi.org/10.1007/978-3-319-46487-9_3)
20. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
21. Liu, D., Li, Q., Dinh, A.D., Jiang, T., Shah, M., Xu, C.: Diffusion action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10139–10149 (October 2023)
22. Liu, J., Wang, X., Wang, C., Gao, Y., Liu, M.: Temporal decoupling graph convolutional network for skeleton-based gesture recognition. *IEEE Trans. Multimedia* **26**, 811–823 (2024)
23. Pandurangan, S., Papandrea, M., Gelsomini, M.: Fine-grained human activity recognition - a new paradigm. In: Proceedings of the 7th International Workshop on Sensor-Based Activity Recognition and Artificial Intelligence. iWOAR '22, Association for Computing Machinery, New York, NY, USA (2023)
24. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
25. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
26. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Trans. Image Process.* **29**, 9532–9545 (2020)
27. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. p. 729–738. UbiComp '13, Association for Computing Machinery, New York, NY, USA (2013)
28. Su, K., Liu, X., Shlizerman, E.: Predict & cluster: Unsupervised skeleton based action recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
29. Wang, C., Zhang, Y., Cui, M., Ren, P., Yang, Y., Xie, X., Hua, X.S., Bao, H., Xu, W.: Active boundary loss for semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2397–2405 (2022)
30. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.: Cross-view action modeling, learning, and recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2649–2656. IEEE Computer Society, Los Alamitos, CA, USA (jun 2014)

31. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7303–7313 (October 2021)
32. Wu, D., Guo, Z., Li, A., Yu, C., Gao, C., Sang, N.: Conditional boundary loss for semantic segmentation. *IEEE Trans. Image Process.* **32**, 3717–3731 (2023)
33. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
34. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI'18/IAAI'18/EAAI'18, AAAI Press (2018)
35. Zhang, J., Tsai, P., Tsai, M.: Semantic2graph: graph-based multi-modal feature fusion for action segmentation in videos. *Appl. Intell.* **54**, 2084–2099 (2024)



# Text-Enhanced Zero-Shot Action Recognition: A Training-Free Approach

Massimo Bosetti<sup>1</sup>(✉), Shibingfeng Zhang<sup>3</sup>, Bendetta Liberatori<sup>1</sup>,  
Giacomo Zara<sup>1</sup>, Elisa Ricci<sup>1,2</sup>, and Paolo Rota<sup>1</sup>

<sup>1</sup> University of Trento, Trento, Italy  
massimo.bosetti@unitn.it

<sup>2</sup> Fondazione Bruno Kessler (FBK), Trento, Italy

<sup>3</sup> University of Bologna, Bologna, Italy

**Abstract.** Vision-language models (VLMs) have demonstrated remarkable performance across various visual tasks, leveraging joint learning of visual and textual representations. While these models excel in zero-shot image tasks, their application to zero-shot video action recognition (ZSVAR) remains challenging due to the dynamic and temporal nature of actions. Existing methods for ZS-VAR typically require extensive training on specific datasets, which can be resource-intensive and may introduce domain biases. In this work, we propose **Text-Enhanced Action Recognition (TEAR)**, a simple approach to ZS-VAR that is training-free and does not require the availability of training data or extensive computational resources. Drawing inspiration from recent findings in vision and language literature, we utilize action descriptors for decomposition and contextual information to enhance zero-shot action recognition. Through experiments on UCF101, HMDB51, and Kinetics-600 datasets, we showcase the effectiveness and applicability of our proposed approach in addressing the challenges of ZS-VAR. (The code will be released later at <https://github.com/MaXDL4Phys/tear>).

**Keywords:** Action Recognition · Zero-shot Transfer · Vision and Language

## 1 Introduction

Multimodal vision-language models (VLMs) [12, 33] have demonstrated outstanding performance across diverse visual tasks. These models undergo pre-training on large-scale datasets, aiming to jointly learn representations for images and text. Benefiting from textual representations, VLMs have exhibited impressive zero-shot capabilities, *i.e.*, ability to generalize to a novel set of unseen classes on a handful of tasks, such as image classification [47], object detection [10, 37]

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_21](https://doi.org/10.1007/978-3-031-78354-8_21).

and segmentation [35]. However, despite the zero-shot transfer results achieved on image tasks, these models still struggle when applied zero-shot to videos without proper fine-tuning [14, 25, 44]. Understanding actions in video streams is inherently more challenging than recognizing static elements in images. For instance, while identifying an object in an image may be straightforward, grasping intricate actions, such as dancing, involves understanding dynamic movements and temporal context, adding complexity to the task. This characteristic makes video action recognition, which finds real-world applications in various fields [6] like autonomous driving, sports analysis, and entertainment, typically more challenging than the image counterpart.

Recognizing actions in videos through zero-shot video action recognition (ZS-VAR) using VLMs can be challenging due to the associated temporal dynamics and complexities. Additional training is often required to capture these factors. Recent ZS-VAR methods have shown satisfactory results but require extensive training on appropriate datasets to achieve such performance. [14, 18, 26, 41]. While effective, these approaches have several drawbacks. Primarily, the training process can be time-consuming and resource-intensive. Additionally, fine-tuning task-specific datasets may introduce biases into the system, limiting its generalizability across different datasets [17]. Furthermore, introducing new parameters can increase the computational cost of model deployment and inference, adding to the complexity of these approaches in real-world scenarios.

These motivations prompt us to explore an alternative approach to ZS-VAR that is training-free and does not require the availability of training data or extensive computational resources. One recently highlighted problem of VLMs is that they may not encode sufficient knowledge of verbs, which are crucial for understanding and recognizing actions in videos [24, 28, 42]. Additionally, research has shown that incorporating contextual information in textual prompts can enhance the performance of VLMs in various downstream tasks [1, 21]. Drawing inspiration from these recent findings, we aim to leverage the decomposition of actions and the introduction of contextual information to improve zero-shot action recognition without further training.

We propose TEAR, which stands for Text-Enhanced Zero-Shot Action Recognition, as a training-free approach for ZS-VAR. We leverage a VLM pre-trained solely on image data, abstaining from fine-tuning it on video data. Our approach unfolds in two primary steps: first, the generation of action descriptors employing a large language model (LLM); second, zero-shot prediction facilitated by the generated textual descriptors. We evaluate the proposed approach on three standard benchmarks, *i.e.*, UCF101 [40], HMDB51 [16], and Kinetics-600 [4].

Our contributions can be summarized as follows:

1. We propose TEAR, Text-Enhanced Action Recognition, the first method addressing zero-shot video action recognition in a training-free manner. Our approach does not rely on the availability of training data or require significant computational resources. This contribution makes ZS-VAR more accessible and practical for real-world applications.

2. By decomposing action labels into sequential observable steps and providing visually related descriptions, our approach enables better understanding and recognition of actions in videos. We demonstrate how leveraging decomposition and description benefits the zero-shot action recognition task.
3. We empirically show the capabilities of the proposed method on three datasets, *i.e.*, UCF101 [40], HMDB51 [16], and Kinetics-600 [4], achieving results that are competitive with training-based approaches.

## 2 Related Work

**Vision-language models.** Vision-language models (VLMs), such as CLIP [33], have been developed to learn joint visual-text embedding spaces through pre-training on large-scale datasets of web-crawled image-text pairs. They have showcased outstanding performance across various downstream tasks, particularly in the image domain, with notable zero-shot capabilities [12]. These models have been recently extended to the video domain, where tasks are typically more challenging due to the additional temporal dimension. Recent works achieve this by incorporating additional learnable components for spatiotemporal modeling, including self-attention layers, textual or vision prompts, or dedicated visual decoders, demonstrating improvements in video-related tasks [14,41]. However, their adaptation to zero-shot settings still necessitates further development, and the results currently lag significantly behind those achieved in tasks related to image processing. Moreover, by introducing new parameters, these methods necessitate additional training and the availability of large-scale training data. This dependence makes the adaptation resource-intensive and can further introduce domain bias, limiting zero-shot transfer on unseen classes.

**Zero-shot action recognition.** Zero-shot action recognition consists of identifying actions in videos from a closed set of action classes not encountered during the model’s training phase. Early work [19,46] proposed to represent actions by sets of manually defined high-level semantic concepts, *i.e.*, attributes, and show that this can be used to recognize action categories that have never been seen before. This advancement represented a step toward more explicit, semantics-driven solutions, as opposed to the modeling of input sequences in latent spaces [7,9,23,30]. Another line of work [2,20,32,39] uses word embedding of action names as semantic representation. Our work differs from these due to the idea of the language modality alone being the key for generalizing to new tasks and categories in a specifically video-oriented fashion.

**Vision-language for action recognition** Previous works have explored the potential of leveraging the newly advanced VLMs, such as CLIP [33], to enhance recognition capabilities with textually conveyed semantics [21,29,38]. These works, however, address the more generic task of image-based recognition without employing text-oriented solutions tailored for videos. On the other hand,

the video field has been investigated in many subsequent works based on video captioning [8] and improved textual descriptors [26, 34, 41, 45, 48]. Our work is more closely aligned with methods using LLMs [11, 18, 34]. MAXI [18] adapts a VLM for zero-shot action recognition using only unlabelled videos, composing a text bag for each unlabelled video using a captioning model and an LLM. FROSTER [11] tackles open-vocabulary action recognition and uses an LLM as a form of text augmentation at training time to mitigate the distribution shift between CLIP’s pre-training captions and template-embedded action names.

However, we move a step further towards specifically video-oriented solutions by employing text-based augmentations to the label space that explicitly leverage the temporal and sequential nature of video data, such as the decomposition of action into sequential sub-actions.

### 3 Method

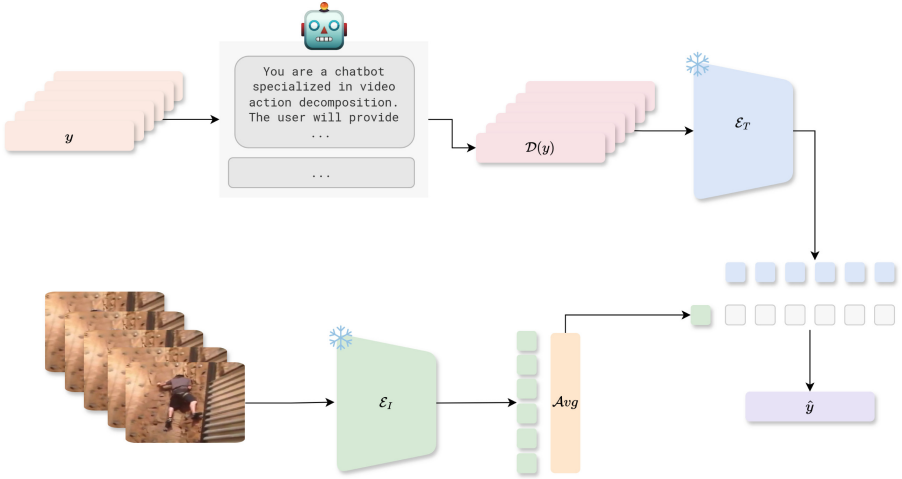
Most prior works in zero-shot video action recognition have focused on adapting image-based VLMs through additional training, necessitating video data availability. In this work, we propose to leverage a language-driven manipulation of action labels and demonstrate that it enables effective action recognition without the need for further training, thereby achieving zero-shot performance. Our proposed method TEAR directly addresses zero-shot action recognition at inference time in a remarkably simple but efficient way, as illustrated in Fig. 1.

Formally, given a video  $\mathcal{V}$  and a pre-defined set of action classes  $\mathcal{C}$ , our goal is to classify the action present in the video. We achieve this with an image-based VLM model and an LLM, without necessitating tailored fine-tuning on video data. TEAR employs a pre-trained CLIP [33] as the VLM and GPT-3.5 [3] for the LLM. The method consists of two main steps: firstly, generating action textual descriptors using a large language model (LLM), and secondly, facilitating zero-shot prediction through these descriptors. We provide detailed explanations of these steps in the following.

#### 3.1 Action descriptors generation

It has been shown that VLMs often struggle with verbs due to their strong object and noun bias [24]. Our key insight is that an action is more than just the verb; the surrounding context and objects can describe it, the different steps needed to perform it and additional visual cues. For each category  $y \in \mathcal{C}$  we construct a set of textual descriptors  $\mathcal{D}(y)$ , considering the following descriptors:

- **Class:** the action label  $y$  in the original format.
- **Decomposition:** a list of sub-actions. Specifically, we break down the action into three consecutive stages, capturing it across different temporal phases.
- **Description:** an elaborate semantic description of the action.



**Fig. 1. Overview of the proposed method.** TEAR addresses the task of zero-shot action recognition. First, for every action class label  $y$ , we generate a set of action textual descriptors  $\mathcal{D}(y)$  by querying an LLM. Then we compute the textual and visual embeddings, keeping both the image and text encoders frozen ( $\text{❄}$ ). Lastly, the final prediction is obtained by computing the similarity between the textual embeddings and the averaged visual embeddings.

- **Context:** a textual descriptor encompassing two distinct types of information pertinent to the action. One is the overall context, highlighting visual features likely to be observed in a video portraying the action. The other is a list of objects likely to participate in the action.
- **Combinations:** a combination of all the previously listed ones.

Crafting textual descriptors for classes manually becomes increasingly impractical as the number of datasets and classes grows, rendering it infeasible. For this reason, we propose to automatically construct this set by prompting a large language model, such as GPT-3.5 [3], with multiple queries. We design a query for each one of the textual descriptors, as reported in Tab. 1. An example of the obtained descriptors for the action *snowboarding* is reported in Tab. 2.

Visual inspection of the obtained  $\mathcal{D}(y)$  against actual video content of the corresponding action class  $y$  confirms the descriptors’ relevance. In Fig. 2 and 3, we illustrate a few examples of action labels, the generated descriptors, and four frames from a video of the same ground truth action. In particular, Fig. 2a, Fig. 2b and Fig. 2c show that: i) the decomposition into steps corresponds to the sub-actions present in the video, describing the whole event in a set of more atomic actions, ii) the description is aligned with the general video content and, iii) the context and objects tags can be found in the video. This approach may result in failure cases when the textual descriptors do not accurately capture specific nuances. In Fig. 3, for example, the obtained descriptors depict a kissing



action in a romantic setting, while a video labeled with the same action portrays a friendly interaction between babies.

Motivated by previous research demonstrating the efficacy of prompt templates [33], we incorporate a diverse set of templates, listed in supplementary material, into our approach. Specifically, we encapsulate all the obtained textual descriptors with the templates. Moreover, prepending the action class for each descriptor typically enhances performance. We attribute this to the fact that omitting the action class altogether and relying solely on the generated descriptors can result in a loss of information. This approach ensures that the generated descriptions maintain relevance and specificity.

### 3.2 Zero-shot recognition with action descriptors

TEAR operates in the following straightforward manner to generate the final inference based on the previously discussed textual descriptors provided by the language model. The key components are a pre-trained VLM, consisting of an image encoder  $\mathcal{E}_I$  and a text encoder  $\mathcal{E}_T$ . Given a test video  $\mathcal{V}$ , first, we sample  $N$  frames uniformly along the whole duration of the video and represent it as a set of frames as  $\mathcal{V} = \{x_i\}_{i=1}^N$ . Then we compute a compact representation from  $\mathcal{V}$  as the average of its  $N$  frames' latent representations, extracted with the vision encoder:

$$\bar{\mathcal{V}} = \frac{1}{N} \sum_{i=1}^N \mathcal{E}_I(x_i) \quad (1)$$

Then we compute a textual representation for each class  $y_j \in \mathcal{C}$ , by encoding the textual descriptors  $\mathcal{D} = \{d_j\}_{j=1}^M$  with the text encoder and averaging them:

$$z_j = \frac{1}{M} \sum_{i=1}^M \mathcal{E}_T(d_i(y_j)) \quad (2)$$

where  $M$  is not fixed and depends on the category  $y_j$ .

Lastly, our model selects the action with the highest cosine similarity to the compact video representation, allowing it to make the final predictions:

$$\hat{y} = \operatorname{argmax}_{j \in |\mathcal{C}|} \left( \frac{z_j \cdot \bar{\mathcal{V}}}{\|z_j\| \cdot \|\bar{\mathcal{V}}\|} \right) \quad (3)$$

## 4 Experiment Results

### 4.1 Datasets and Metrics

We conduct experiments with three popular video action recognition datasets: UCF101 [40], HMDB51 [16], and Kinetics-600 (K600) [4]. These datasets are frequently used to evaluate zero-shot action recognition. We report the standard

## (a) "making pizza"

Description: A person kneads dough, spreads tomato sauce, sprinkles cheese, and adds toppings like pepperoni, vegetables, and herbs on a pizza crust, then bakes it in an oven until the cheese melts and the crust turns crispy, creating a delicious homemade pizza."

Step 1: "Stretch or roll out pizza dough on a floured surface",

Step 2: "Spread tomato sauce evenly over the dough",

Step 3: "Top with cheese and desired toppings, then bake in a preheated oven until crust is golden and cheese is bubbly"

Context: "a kitchen or pizzeria",  
"dough",  
"tomato sauce",  
"cheese",  
"oven"



## (b) "bowling"

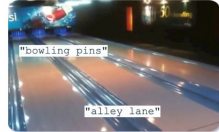
Description: A person stands at the end of a bowling lane, holds a bowling ball and approaches the foul line, steps forward and swings the ball to release it down the lane, aiming to knock down standing pins, showcasing technique, precision, and accuracy in the sport of bowling.

Step 1: "Select a suitable bowling ball and position yourself at the start of the approach"

Step 2: "Align your body towards the pins, holding the ball with both hands"

Step 3: "Take a few steps forward, swing the ball back and then forward, releasing it onto the lane towards the pins"

Context: "a bowling alley"  
"bowling ball"  
"bowling pins"  
"bowling shoes"  
"alley lane"



## (c) "looking phone"

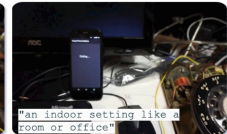
Description: A person holds a mobile device in their hand, gazes at the screen, scrolls through content, reads messages, watches videos, or engages with apps, while tapping, swiping, or interacting with the touchscreen to access information or communicate digitally.

Step 1: "Unlock your phone by pressing the power button or using biometrics"

Step 2: "Hold the phone in front of you at a comfortable distance"

Step 3: "Use your fingers to navigate the screen, opening apps or scrolling through content"

Context: "an indoor setting like a room or office"  
"phone"  
"person"  
"screen"



**Fig. 2. Examples of descriptors matching visual cues in test videos.** We show descriptors generated for four videos of Kinetics-600. We show four frames for each video and highlight the matching with the decomposition, description, and context. For each video, the label above represents the ground truth label.



**Fig. 3. Example of descriptors that do not match visual cues in test videos.** We show descriptors generated for one video of Kinetics-600 of the class `kissing`. We show four frames from the video and highlight the matching with the decomposition, description, and context. For this sample, the textual descriptors do not match the visual cues in the video. Further qualitative analyses are available in the supplementary material.

evaluation metrics of Top1/Top5 accuracy. To ensure our experiments are comparable to previous studies, we adopt the same protocol of previous works [18, 36].

The HMDB51 dataset [16] contains approximately 7,000 manually annotated videos of human motion sourced from various platforms, including films and YouTube. Each video is categorized under one of 51 action labels, with at least 101 videos per label. The average duration of each video is 3.2 seconds.

**UCF101.** The UCF101 dataset [40] consists of 13,320 videos derived from various online platforms and categorized into 101 action classes. These classes encompass a wide range of human activities and are organized into five broad categories: Human Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports.

**Kinetics-600.** Extending the Kinetics-400 dataset [15], Kinetics-600 [4] features videos representing 600 human action classes. The additional videos, sourced from YouTube, broaden the range of depicted actions to include various interpersonal and person-object interactions and individual actions.

## 4.2 Implementation details

We extract RGB frames and resize them to a resolution of  $224 \times 224$ . We employ CLIP (with ViT-B/16 visual encoder) as the VLM and GPT-3.5 as the LLM. We do not provide details on training implementation, as our proposed TEAR is inference-only. The sole hyperparameter, the number of frames sampled from the video ( $N = 16$ ), is set to align with state-of-the-art methods. The number of textual descriptors per-class  $M$  varies among different classes, as we do not set it a priori and depends on the output of the LLM.

**Table 1. Queries used for action description generation.** We show the prompts used to query the LLM for each textual descriptor generation.

Descriptor	Query
Decomposition	You are a chatbot specialised in video action decomposition. The user will provide you with an action and you will have to decompose it into three sequential observable steps. The steps must strictly be three. You must strictly provide each response as a python list, e.g., ['action1', 'action2', action3']. Omit any kind of introduction, the response must only contain the three actions. Comply strictly to the template. Do not ask for any clarification, just give your best answer. It is for a school project, so it is very important. It is also very important your response is in the form of a python list.
Description	You are a chatbot specialised in video action description. The user will provide you with an action and you will have to describe the action by providing only visually related information. You must strictly provide each response as a Python string. The description should be succinct and general. Omit any kind of introduction. Comply strictly to the template. Do not ask for any clarification, just give your best answer. Following is an example. Action label: typing. Description: Typing normally involve a person and a device with keyboard. When typing, the individual positions their fingers over the keyboard.
Context	You are a chatbot specialised in video understanding. The user will provide you with the name of an action, and you will have to provide two specific pieces of information about that action. The first one is the context, which consists of any visually relevant feature that may be expected to appear in a video portraying that action. The second one consists of a lists of objects that may involved in the action. You must strictly provide each response as a python dictionary, e.g., context: person, objects: [person]. Omit any kind of introduction, the response must only contain the two pieces of information. Comply strictly to the template. Do not ask for any clarification, just give your best answer. It is for a school project, so it is very important. It is also very important your response is in the form of a python dictionary.

### 4.3 Comparative results

In Tab. 3, it can be seen that vanilla CLIP already has good zero-shot performance across the three datasets. It outperforms training-based methods like ER-ZSAR [5] and JigsawNet [31] without fine-tuning on video data. The remaining training-based methods adapt CLIP by fine-tuning on Kinetics-400. Most of these approaches are supervised, while MAXI [18] and LSS [34] perform fine-tuning on an unlabeled video data collection. With TEAR, we eliminate the need for training, enabling direct inference. On UCF101 and HMDB51, our results significantly surpass the CLIP baseline, achieving +6.3% and +12.8% Top 1

**Table 2. Example of generated action description.** We show an example for the specific action of *snowboarding*.

Descriptor	Content
<b>Class</b>	"snowboarding"
<b>Decomposition</b>	"Strap your feet securely onto the snowboard bindings", "Lean forward to initiate movement down the slope", "Use heel-to-toe shifts in weight to steer and balance as you descend"
<b>Description</b>	"A person sliding down a snow-covered slope on a single board attached to their feet, making turns and jumps while maintaining balance."
<b>Context</b>	"snow-covered mountain slope or snow park", "snowboard", "snow boots", "helmet"
<b>Combination</b>	"snowboarding", "Strap your feet securely onto the snowboard bindings", "Lean forward to initiate movement down the slope", "Use heel-to-toe shifts in weight to steer and balance as you descend", "A person sliding down a snow-covered slope on a single board attached to their feet, making turns and jumps while maintaining balance.", "snow-covered mountain slope or snow park", "snowboard", "snow boots", "helmet"

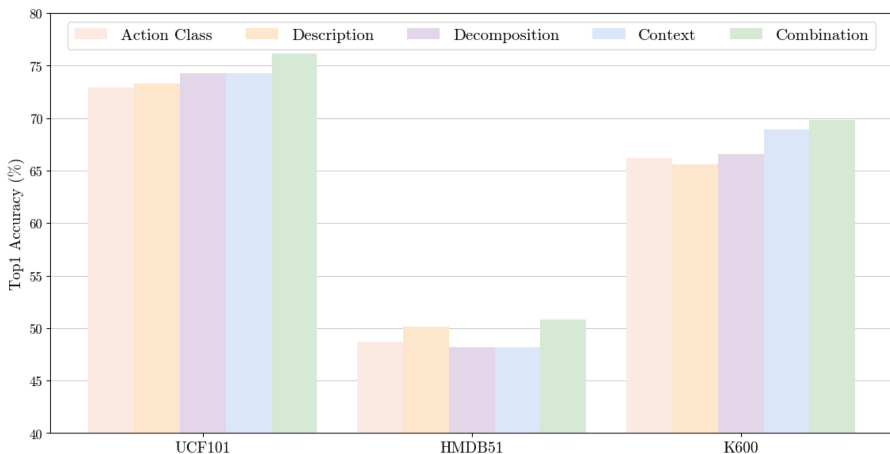
accuracy, respectively. Additionally, on Kinetics-600, TEAR improves upon the baseline (+6.8/5.3% Top1/Top5).

#### 4.4 Ablation

In this section, we perform ablations of our method to validate our main design choices. We report Top1/Top5 accuracy for all of the datasets considered. In the ablation shown in Fig. 4, we evaluate the choice of the textual descriptors used in the proposed methodology, as detailed in Sec. 3.1. Our findings indicate that incorporating one of the descriptors usually enhances performance. However, the most substantial improvement is observed when all the descriptors are used in conjunction. Hence, a comprehensive approach furnishes the VLM model with richer linguistic cues, enhancing its zero-shot action recognition accuracy across all benchmarks. In Tab. 4, we depict the results of the ablations of different descriptors, discussed in detail in supplementary material, the use of additional templates and the choice of prepending the original action class to the obtained textual descriptors. We observe that adding the original class label and using templates enhances the model's accuracy.

**Table 3. Comparison with state-of-the-art zero-shot action recognition methods.** We report zero-shot action recognition results on UCF101, HMDB51, and K600. We report Top1 and Top5 accuracy computed on the three official test splits. We also include the backbone used and the number of frames sampled from videos. The green color is **our method**.

Method	Training	Backbone	Frames	UCF101		HMDB51		K600	
				Top1	Top5	Top1	Top5	Top1	Top5
ER-ZSAR [5]	✓	TSM	16	51.8	35.3	42.1	73.1	-	-
JigsawNet [31]	✓	R(2+1)D	16	56.0	38.7	-	-	-	-
ActionCLIP [41]	✓	ViT-B/16	32	58.3	40.8	66.7	91.6	-	-
XCLIP [26]	✓	ViT-B/16	32	72.0	44.6	65.2	86.1	-	-
A5 [14]	✓	ViT-B/16	32	69.3	44.3	55.8	81.4	-	-
ViFi-CLIP [36]	✓	ViT-B/16	32	76.8	51.3	71.2	92.2	-	-
Text4Vis [43]	✓	ViT-L/14	16	-	-	68.9	-	-	-
MAXI [18]	✓	ViT-B/16	16/32	78.2	52.3	<b>71.5</b>	<b>92.5</b>	-	-
LSS [34]	✓	ViT-B/16	8	74.2	51.4	-	-	-	-
OTI [48]	✓	ViT-B/16	8	<b>88.3</b>	<b>54.2</b>	-	-	-	-
EPK-CLIP[45]	✓	ViT-B/16	8	75.3	48.7	-	-	-	-
EPK-ViFi[45]	✓	ViT-B/16	8	77.7	51.6	-	-	-	-
CLIP [33]	✗	ViT-B/16	16	69.9	38.0	63.5	86.8	-	-
TEAR	✗	ViT-B/16	16	76.2	50.8	70.3	92.1	-	-



**Fig. 4. Ablation on using the textual descriptors.** We ablate the use of different textual descriptors defined in Sec. 3.1. We report the Top1 accuracy on the three datasets and use the same color coding as in Sec. 3.1.

**Table 4. Ablation on constructing the textual prompts.** We ablate using templates and prepending the action class after descriptor generation. Results are reported for both ViT-B/32 and ViT-B/16 visual backbones. Green is **our configuration**.

Template	Class	Backbone	UCF101		HMDB51		K600	
			Top1	Top5	Top1	Top5	Top1	Top5
✗	✗	ViT-B/32	68.9	93.1	43.6	73.9	61.1	86.9
✓	✗	ViT-B/32	68.7	92.8	45.8	74.1	61.5	87.4
✗	✓	ViT-B/32	72.2	94.0	47.5	76.9	66.9	89.5
✓	✓	ViT-B/32	72.6	94.6	48.0	78.5	67.0	90.0
✗	✗	ViT-B/16	70.1	94.0	44.1	75.6	66.0	90.5
✓	✗	ViT-B/16	72.6	94.6	48.4	77.0	66.2	90.1
✗	✓	ViT-B/16	75.8	95.9	49.0	81.2	70.2	92.2
✓	✓	ViT-B/16	<b>76.2</b>	<b>96.3</b>	<b>50.8</b>	<b>82.0</b>	<b>70.3</b>	<b>92.3</b>

In addition, in Tab. 5, we assess the choice of the visual backbone  $\mathcal{E}_I$  and the number of sampled frames  $N$ . We observe a significant gain with ViT-B/16 compared to ViT-B/32, and ViT-B/16 is also the backbone commonly employed by other competitors. Additionally, our method exhibits low sensitivity to the number of sampled frames for both backbones, whether 16 or 32. As a result, we adopt the configuration with 16 sampled frames as our final choice to have a fair comparison with most competitors who also utilize this setting.

**Table 5. Ablation on the backbone used and the frame sampling  $N$ .** Green is **our configuration**.

Backbone	N	UCF101		HMDB51		K600	
		Top1	Top5	Top1	Top5	Top1	Top5
ViT-B/32	32	72.5	94.7	48.5	78.2	67.0	90.0
ViT-B/32	16	72.2	94.0	47.5	76.9	66.8	89.9
ViT-B/16	32	76.4	96.5	50.8	82.2	70.5	92.2
ViT-B/16	16	<b>76.2</b>	<b>96.3</b>	<b>50.8</b>	<b>82.0</b>	<b>70.3</b>	<b>92.3</b>

Lastly, we ablate the different LLMs to determine the robustness of the method related to the generation of action descriptors. Thus, we re-evaluate our method using different LLMs on the HMDB dataset.

Tab. 6 revealed that our method is robust, with only minor performance differences across different LLMs. Although advanced models like GPT-4o offer

**Table 6. Ablation on the LLM used to generate prompts.** Green is our configuration .

LLM	Description		Decomposition		Context		Combination	
	Top1	Top5	Top1	Top5	Top1	Top5	Top1	Top5
GPT-3.5	50.1	79.2	48.2	81.1	48.2	81.1	50.8	82.0
GPT-4o [27]	51.4	81.3	49.7	79.8	49.9	80.1	50.8	82.3
Llama3 [22]	49.5	80.2	49.1	77.9	49.5	79.4	49.5	80.1
Mistral [13]	47.8	78.3	47.4	78.9	47.8	79.8	45.2	74.8

slight improvements, our method remains effective regardless of the model used. This showcases the method’s reliability and adaptability to various LLMs with varying capacities. We maintain the use of GPT-3.5 as it offers a cost-effective alternative to GPT-4o, ensuring the method remains accessible without sacrificing significant performance.

In conclusion, our experiments, which combine all forms of text augmentation—including label, description, decomposition, and context—significantly when templates and class label conditioning are applied, demonstrate a cumulative improvement in performance.

## 5 Limitations

While our approach for generating textual action descriptors provides an automated pipeline for capturing various aspects of action classes, it may be sub-optimal for more temporally fine-grained or very atomic actions that cannot be decomposed into distinct steps. Additionally, our method may encounter limitations when dealing with actions that exhibit less association with objects or are highly variable in context. Actions with weaker object associations may benefit less from the generated textual descriptors. Similarly, actions that vary widely in context may result in descriptors that fail to capture the diverse contexts in which they occur. Addressing these limitations can lead to more advanced models for language-driven action recognition.

## 6 Conclusion

This work tackles the challenging problem of zero-shot video action recognition. We propose TEAR, a training-free approach that generates rich textual descriptors for the action class labels and then performs zero-shot prediction using the obtained descriptors. Despite its simplicity, TEAR outperforms baseline models and rivals training-based methods in the task of zero-shot action recognition, all without the need for in-domain training.



While our method was primarily evaluated on action recognition, its applicability can extend to more challenging tasks on untrimmed videos, such as Temporal Action Localization or Action Segmentation. By leveraging textual descriptors to bridge the semantic gap between action labels and visual content, our approach promises to tackle broader video understanding tasks beyond mere classification. Future research efforts should explore the adaptation and extension of TEAR to address these more complex video analysis tasks.

**Acknowledgements.** We acknowledge the PRECRISIS project, funded by the EU Internal Security Fund (ISFP-2022-TFI-AG-PROTECT-02-101100539), MUR PNRR project FAIR - Future AI Research (PE00000013), funded by NextGeneration EU and the CINECA award under the ISCRA initiative for the availability of high-performance computing resources and support. We also thank the Deep Learning Lab of the ProM Facility for the GPU time.

## References

1. An, B., Zhu, S., Panaitescu-Liess, M.A., Mummadi, C.K., Huang, F.: More context, less distraction: Visual classification by inferring and conditioning on contextual attributes. *arXiv* (2023)
2. Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., Chalupka, K.: Rethinking zero-shot video classification: End-to-end training for realistic applications. In: *CVPR* (2020)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *NeurIPS* (2020)
4. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. *arXiv* (2018)
5. Chen, S., Huang, D.: Elaborative rehearsal for zero-shot action recognition. In: *ICCV* (2021)
6. Deng, A., Yang, T., Chen, C.: A large-scale study of spatiotemporal representation learning with a new benchmark on action recognition. In: *ICCV* (2023)
7. Doshi, K., Yilmaz, Y.: Zero-shot action recognition with transformer-based video semantic embedding. In: *CVPRW* (2023)
8. Estevam, Laroca, P.e.a.: Tell me what you see: A zero-shot action recognition method based on natural language descriptions. In: *Multimed Tools Appl* (2024)
9. Gao, J., Hou, Y., Guo, Z., Zheng, H.: Learning spatio-temporal semantics and cluster relation for zero-shot action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* (2023)
10. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. In: *ICLR* (2022)
11. Huang, X., Zhou, H., Yao, K., Han, K.: FROSTER: Frozen CLIP is a strong teacher for open-vocabulary action recognition. In: *ICLR* (2024)
12. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *ICML* (2021)
13. Jiang, A., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b (2023). *arXiv* (2023)

14. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV (2022)
15. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv (2017)
16. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: A large video database for human motion recognition. In: ICCV (2011)
17. Liberatori, B., Conti, A., Rota, P., Wang, Y., Ricci, E.: Test-time zero-shot temporal action localization. arXiv (2024)
18. Lin, W., Karlinsky, L., Shvetsova, N., Possegger, H., Kozinski, M., Panda, R., Feris, R., Kuehne, H., Bischof, H.: Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In: ICCV (2023)
19. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
20. Mandal, D., Narayan, S., Dwivedi, S.K., Gupta, V., Ahmed, S., Khan, F.S., Shao, L.: Out-of-distribution detection for generalized zero-shot action recognition. In: CVPR (2019)
21. Menon, S., Vondrick, C.: Visual classification via description from large language models. In: ICLR (2023)
22. Meta, A.: Introducing meta llama 3: The most capable openly available llm to date. Meta AI (2024)
23. Mettes, P.: Universal prototype transport for zero-shot action recognition and localization. IJCV (2023)
24. Momeni, L., Caron, M., Nagrani, A., Zisserman, A., Schmid, C.: Verbs in action: Improving verb understanding in video-language models. In: ICCV (2023)
25. Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Zero-shot temporal action detection via vision-language prompting. In: ECCV (2022)
26. Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., Ling, H.: Expanding language-image pretrained models for general video recognition. In: ECCV (2022)
27. OpenAI: Chatgpt: Gpt-4 (2024), <https://www.openai.com/>, accessed: 2024-07-05
28. Park, J.S., Shen, S., Farhadi, A., Darrell, T., Choi, Y., Rohrbach, A.: Exposing the limits of video-text models through contrast sets. In: ACL (2022)
29. Pratt, S., Covert, I., Liu, R., Farhadi, A.: What does a platypus look like? generating customized prompts for zero-shot image classification. In: ICCV (2023)
30. Qi, C., Feng, Z., Xing, M., Su, Y., Zheng, J., Zhang, Y.: Energy-based temporal summarized attentive network for zero-shot action recognition. IEEE Transactions on Multimedia (2023)
31. Qian, Y., Yu, L., Liu, W., Hauptmann, A.G.: Rethinking zero-shot action recognition: Learning from latent atomic actions. In: ECCV (2022)
32. Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., Wang, Y.: Zero-shot action recognition with error-correcting output codes. In: CVPR (2017)
33. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
34. Ranasinghe, K., Ryoo, M.S.: Language-based action concept spaces improve video self-supervised learning. In: NeurIPS (2024)
35. Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., Lu, J.: Denseclip: Language-guided dense prediction with context-aware prompting. In: CVPR (2022)

36. Rasheed, H., Khattak, M.U., Maaz, M., Khan, S., Khan, F.S.: Fine-tuned clip models are efficient video learners. In: CVPR (2023)
37. Rasheed, H., Maaz, M., Khattak, M.U., Khan, S., Khan, F.S.: Bridging the gap between object and image-level representations for open-vocabulary detection. In: NeurIPS (2022)
38. Roth, K., Kim, J.M., Koepke, A.S., Vinyals, O., Schmid, C., Akata, Z.: Waffling around for performance: Visual classification with random words and broad concepts. In: ICCV (2023)
39. Shao, H., Qian, S., Liu, Y.: Temporal interlacing network. In: AAAI (2020)
40. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv (2012)
41. Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv (2021)
42. Wang, Z., Blume, A., Li, S., Liu, G., Cho, J., Tang, Z., Bansal, M., Ji, H.: Paxion: Patching action knowledge in video-language foundation models. NeurIPS (2024)
43. Wu, W., Sun, Z., Ouyang, W.: Revisiting classifier: Transferring vision-language models for video recognition. In: AAAI (2023)
44. Yan, S., Xiong, X., Nagrani, A., Arnab, A., Wang, Z., Ge, W., Ross, D., Schmid, C.: Unloc: A unified framework for video localization tasks. In: ICCV (2023)
45. Yang, Z., An, G., Zheng, Z., Cao, S., Wang, F.: Epk-clip: External and priori knowledge clip for action recognition. Expert Systems with Applications (2024)
46. Zellers, R., Choi, Y.: Zero-shot activity recognition with verb attribute induction. In: EMNLP (2017)
47. Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free clip-adapter for better vision-language modeling. arXiv (2021)
48. Zhu, Y., Zhuo, J., Ma, B., Geng, J., Wei, X., Wei, X., Wang, S.: Orthogonal temporal interpolation for zero-shot video recognition. In: ACMM-MM (2023)



# ActNetFormer: Transformer-ResNet Hybrid Method for Semi-supervised Action Recognition in Videos

Sharana Dharshikgan Suresh Dass<sup>1</sup>, Hrishav Bakul Barua<sup>1,2</sup>,  
Ganesh Krishnasamy<sup>1</sup>✉, Raveendran Paramesran<sup>1</sup>, and Raphaël C.-W. Phan<sup>1</sup>

<sup>1</sup> School of Information Technology, Monash University, Subang Jaya, Malaysia  
{sharana.sureshdass, hrishav.barua, ganesh.krishnasamy,  
raveendran.paramesran, raphael.phan}@monash.edu

<sup>2</sup> Robotics and Autonomous Systems Lab, TCS Research, Kolkata, India

**Abstract.** Human action or activity recognition in videos is a fundamental task in computer vision with applications in surveillance and monitoring, self-driving cars, sports analytics, human-robot interaction and many more. Traditional supervised methods require large annotated datasets for training, which are expensive and time-consuming to acquire. This work proposes a novel approach using cross-architecture pseudo-labeling with contrastive learning for semi-supervised action recognition. Our framework leverages both labeled and unlabeled data to robustly learn action representations in videos, combining pseudo-labeling with contrastive learning for effective learning from both types of samples. We introduce a novel cross-architecture approach where 3D Convolutional Neural Networks (3D CNNs) and video transformers (VIT) are utilized to capture different aspects of action representations; hence we call it *ActNetFormer*. The 3D CNNs excel at capturing spatial features and local dependencies in the temporal domain, while VIT excels at capturing long-range dependencies across frames. By integrating these complementary architectures within the ActNetFormer framework, our approach can effectively capture both local and global contextual information of an action. This comprehensive representation learning enables the model to achieve better performance in semi-supervised action recognition tasks by leveraging the strengths of each of these architectures. Experimental results on standard action recognition datasets demonstrate that our approach performs better than the existing methods, achieving state-of-the-art performance with only a fraction of labeled data. The official website of this work is available at: <https://github.com/rana2149/ActNetFormer>.

**Keywords:** Video action recognition · Convolutional neural network · Video transformer · Contrastive learning · Deep learning

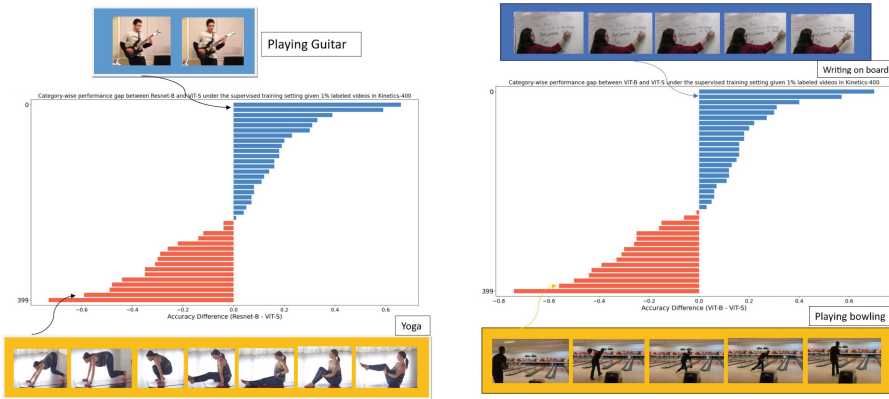
This research is supported by the Global Research Excellence Scholarship, Monash University, Malaysia. This research is also supported, in part, by the Global Excellence and Mobility Scholarship (GEMS), Monash University, Malaysia & Australia.

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78354-8\\_22](https://doi.org/10.1007/978-3-031-78354-8_22).

# 1 Introduction

The remarkable advancements in deep learning have revolutionized action recognition, particularly with the advent of supervised learning protocols. However, acquiring a substantial number of annotated videos remains a challenge in practice since it is time-consuming and expensive [17,39]. Each day, video-sharing platforms like YouTube and Instagram witness millions of new video uploads. Leveraging this vast pool of unlabeled videos presents a significant opportunity for semi-supervised learning approaches, promising substantial benefits for advancing action recognition capabilities [20,37].

A typical method for leveraging unlabeled data involves assigning pseudo-labels to them and effectively treating them as *ground truth* during training [12,22,30]. Current methodologies typically involve training a model on annotated data and subsequently employing it to make predictions on unlabeled videos. When predictions exhibit high confidence levels, they are adopted as pseudo-labels for the respective videos, guiding further network training. However, the efficacy of this approach hugely depends on the quantity and accuracy of the pseudo-labels generated. Unfortunately, the inherent limitations in discriminating patterns from a scant amount of labeled data often result in subpar pseudo-labels, ultimately impeding the potential benefits gleaned from unlabeled data.



(a) The difference in performance between ResNet-B and VIT-S, categorized by class, is evaluated under a supervised training scenario with only 1% labeled videos in the Kinetics-400 dataset.

(b) The difference in performance between VIT-B and VIT-S, categorized by class, is evaluated under a supervised training scenario with only 1% labeled videos in the Kinetics-400 dataset.

**Fig. 1.** Comparison of performance between different architectural models.

To enhance the utilization of unlabeled videos, our approach draws inspiration from recent studies, particularly from [34], which introduced an auxiliary model to provide complementary learning. We also introduce complementary learning but with notable

advancements. Firstly, we introduce a cross-architecture strategy, leveraging both 3D CNNs and transformer models' strengths, unlike CMPL [34], which relies solely on 3D CNNs. This is because both 3D CNNs and video transformers (ViT) offer distinct advantages in action recognition. As shown in Fig. 1a, videos for activities such as 'playing the guitar' from the Kinetics-400 dataset that demonstrate short-range temporal dependencies typically involve actions or events that occur over a relatively short duration and require capturing temporal context within a limited time-frame, and perform better with 3D CNNs. This is because 3D CNNs excel at capturing spatial features and local dependencies in the temporal domain due to their intrinsic property, which involves processing spatio-temporal information through convolutions.

On the other hand, transformer architectures, leveraging self-attention mechanisms, can naturally capture long-range dependencies by allowing each token to learn attention across the entire sequence. As shown in Fig. 1a videos such as the "yoga" class in the Kinetics-400 dataset, which demonstrate long-range temporal dependencies involving actions or events that unfold gradually over extended periods, perform better in the transformer model. Such intrinsic property in transformers enables them to capture complex relationships and interactions between distant frames, leading to a more holistic understanding of the action context. This capability enables transformers to encode meaningful context information into video representations, facilitating a deeper understanding of the temporal dynamics and interactions within the video sequence.

Besides that, CMPL [34] also suggests that smaller models excel at capturing temporal dynamics in action recognition. In comparison, larger models are more adept at learning spatial semantics to differentiate between various action instances. Motivated by this approach, we chose to leverage the advantages of a smaller transformer model, ViT-S, over its larger counterpart, ViT-B. As depicted in Fig. 1b and further studied in Section S2 in the *Supplementary Material*, a smaller model, despite its smaller capacity, does obtain significant improvements over a bigger model in certain classes. While ViT-B excels at capturing spatial semantics, it is essential to note that our primary model, 3D-ResNet50, already possesses these strong capabilities. The 3D convolutional nature of ResNet-50 makes it well-suited for extracting spatial features and local dependencies within the temporal domain. Therefore, the inclusion of ViT-S as an auxiliary model complements the strengths of our primary model by focusing on capturing temporal dynamics, which aligns with our primary objective of addressing action recognition in videos. This strategic combination allows our ActNetFormer framework to achieve a balanced representation learning, leveraging the spatial semantics captured by 3D-ResNet50 and the temporal dynamics captured by ViT-S. As demonstrated in our ablation study (Section 7.2), this integration of ViT-S as an auxiliary model consistently leads to better results compared to adapting ViT-B. Hence, while ViT-B remains essential, its role is effectively supported by the capabilities of our primary model, thereby justifying our choice of prioritizing ViT-S within the ActNetFormer framework.

Furthermore, our method also incorporates video level contrastive learning, enabling the model to glean stronger representations at the spatio-temporal level. Hence, our cross-architecture pseudo-labeling approach is utilized to capture distinct aspects of action representation from both the 3D CNNs and transformer architectures, while

cross-architecture contrastive learning aims explicitly to align the representations and discover mutual information in global high-level representations across these architectures. More experimental details about the cross-architecture strategy are included in Section S1.1 in the *Supplementary Material*.

The main contributions of this work is twofold and listed as follows:

- We propose a novel cross-architecture pseudo-labeling framework for semi-supervised action recognition in videos.
- An architecture-level contrastive learning is developed to enhance the performance of the proposed approach for action recognition in videos.

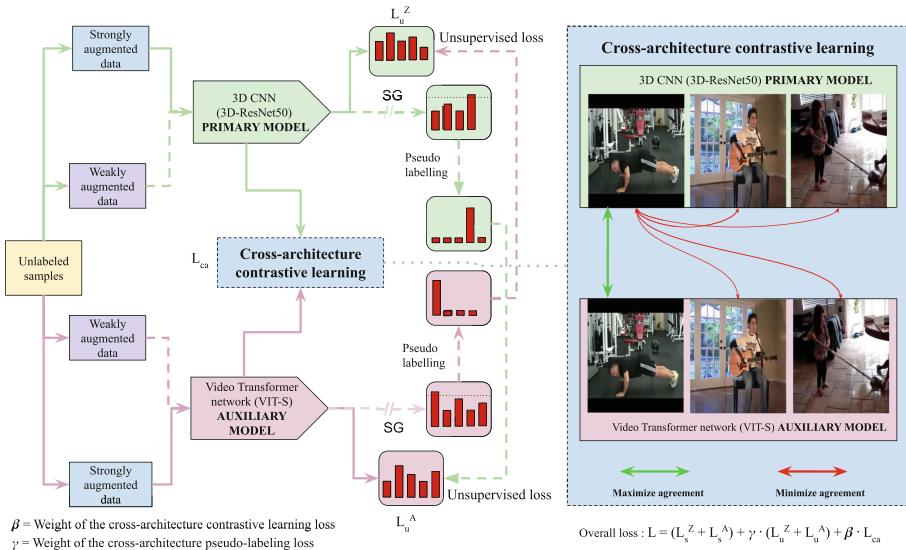


Fig. 2. Architecture of the proposed framework.

## 2 Related works

### 2.1 Action Recognition

Action recognition has advanced significantly with deep learning architectures like CNNs, Recurrent Neural Networks (RNNs), and Transformers. CNNs capture spatial information, while the RNNs captures temporal dependencies [3]. Meanwhile, Transformers, known for NLP tasks, is excellent at capturing long-range dependencies. Varshney *et al.* [28] proposed a CNN model combining spatial and temporal information using different fusion schemes for human activities recognition video. Vision Transformer (ViT) [8] treats images as sequences of patches, achieving competitive

performance on image classification tasks. Arnab *et al.* [1] extend Transformers to video classification, while Bertasius *et al.* [2] introduce TimeSformer, a convolution-free approach to video classification built exclusively on self-attention over space and time convolution-free approach. TimeSformer achieves state-of-the-art (SOTA) results on action recognition benchmarks like Kinetics-400 and Kinetics-600, offering faster training and higher efficiency. Besides that, TimeSformer can also achieve good results even without pretraining. However, achieving these results may require more extensive data augmentation and longer training periods.

## 2.2 Semi-supervised Learning for Video Action Recognition

Action recognition in computer vision is vital across various applications, yet it often suffers from limited labeled data. Semi-supervised learning (SSL) methods provide a solution by utilizing both labeled and unlabeled data to enhance model performance [22, 34]. These approaches exploit the abundance of unlabeled video data available online. Wu *et al.* [31] proposed NCCL, a neighbor-guided consistent and contrastive learning (NCCL) method for semi-supervised video-based action recognition. Xu *et al.* [34] introduced CMPL, employing cross-model predictions to generate pseudo-labels and improve model performance. Singh *et al.* [21] leverage unsupervised videos played at different speeds to address limited labeled data. Xiao *et al.* [32] enhance semi-supervised video action recognition by incorporating temporal gradient information alongside RGB data. Jing *et al.* [13] use pseudo-labels from CNN confidences and normalized probabilities to guide training, achieving impressive results with minimal labeled data. Gao *et al.* [10] introduced an end-to-end semi-supervised Differentiated Auxiliary guided Network (DANet) for action recognition. Xiong *et al.* [33] introduce multi-view pseudo-labeling, leveraging appearance and motion cues for improved SSL. Tong *et al.* [26] propose TACL, employing temporal action augmentation, action consistency learning, and action curriculum pseudo-labeling for enhanced SSL. These advancements demonstrate the potential of SSL techniques in boosting action recognition performance, especially in scenarios with limited labeled data.

## 2.3 Contrastive Learning in Action Recognition

Contrastive learning has become a popular approach, especially in computer vision [16]. Unlike supervised methods, contrastive learning operates on unlabeled data, maximizing agreement between similar samples while minimizing it between dissimilar ones [19, 24]. It fosters a feature space where similar instances are clustered and dissimilar ones are separated. By optimizing a similarity metric using positive (similar) and negative (dissimilar) sample pairs, contrastive learning extracts meaningful features beneficial for tasks like classification and object detection. Its advantage lies in learning from vast unlabeled data, making it suitable for scenarios with limited labeled data [7, 25]. Guo *et al.* [11] propose AimCLR, a contrastive learning-based self-supervised action representation framework. They enhance positive sample diversity and minimize distribution divergence, achieving superior performance. The method in [38] also proposes a hierarchical matching model for few-shot action recognition,



leveraging contrastive learning to enhance video similarity measurements across multiple levels. Rao et al. [18] introduce AS-CAL, a contrastive learning method for action recognition with 3D skeleton data, capturing action patterns across transformations for effective representation.

### 3 Method

#### 3.1 Overview of our work

The proposed ActNetFormer framework is illustrated in Fig. 2. Our approach consists of two models, *i.e.*, the primary model  $Z(\cdot)$  and the auxiliary model  $A(\cdot)$ . These models process video inputs with varying frame rates, utilizing 3D-ResNet50 as the primary model and VIT-S as the auxiliary model by default. When presented with an unlabeled video, both models independently generate predictions on the data that are weakly augmented. The predicted outcomes are then utilized to generate a pseudo-label for the counterpart model, acting as guidance for the strongly augmented data. The “SG” notation denotes the stop-gradient operation, and supervised losses from labeled data are not depicted in this figure. Additionally, we incorporate contrastive learning to maximize agreement between the outputs of the two architectures for the same video while minimizing the agreement for different videos. ActNetFormer leverages the strengths of both a 3D CNN and a transformer. Given an input video clip, each model produces a video representation separately. This encourages each model to focus on different features or patterns within the videos, leading to more comprehensive representations. By combining these complementary representations through contrastive learning, the framework can leverage a richer set of features for action recognition.

#### 3.2 Our proposed framework

Given a labeled dataset  $X$  containing  $N_l$  videos, each paired with a corresponding label  $(x_i, y_i)$ , and an unlabeled dataset  $U$  comprised of  $N_u$  videos, ActNetFormer efficiently learns an action recognition model by utilizing both data that are labeled and unlabeled. Typically, the size of the unlabeled dataset  $N_u$  is greater than that of the labeled dataset  $N_l$ . We provide a brief description of the pseudo-labeling method in Section 3.3. Subsequently, we introduce the proposed ActNetFormer framework in Section 3.4. Then, we explain how contrastive learning works in ActNetFormer framework in Section 3.4. Subsequently, we delve into the implementation details of ActNetFormer in Section 4.

#### 3.3 Preliminaries on Pseudo-Labeling

Pseudo-labeling is a widely employed approach in semi-supervised image recognition, aiming to leverage the model to generate artificial labels for data that are not labeled [22, 36, 39]. The generated labels that surpass a predefined threshold are kept, enabling the associated unlabeled data to be utilized as extra samples for training. Fix-Match [22], a recent SOTA approach, utilizes weakly augmented images for acquiring pseudo-labels, which are subsequently combined with strongly augmented versions to

generate labeled samples. The extension of FixMatch to semi-supervised action recognition can be accomplished as follows:

$$L_u = \frac{1}{B_u} \sum_{i=1}^{B_u} 1(\max(q_i) \geq \tau) \mathcal{H}(\hat{y}_i, Z(G_s(u_i))), \quad (1)$$

In the equation (1),  $B_u$  denotes the batch size,  $\tau$  is the threshold used to indicate if the prediction that is made is reliable or not,  $1(\cdot)$  denotes the indicator function,  $q_i = Z(G_w(u_i))$  represents the class distribution, and  $\hat{y}_i = \arg \max(q_i)$  denotes the pseudo-label.  $G_s(\cdot)$  and  $G_w(\cdot)$  respectively denote the processes of strong and weak augmentation.  $\mathcal{H}(\cdot, \cdot)$  represents the standard cross-entropy loss.  $L_u$  represents the loss on the unlabeled data, while the loss on the labeled data is the cross-entropy loss typically used in action recognition.

### 3.4 Cross-Architecture Pseudo-Labeling

In Section 3.3, we discussed the fundamental concept underlying recent semi-supervised learning methodologies, which revolves around generating high-quality pseudo-labels for unlabeled data. However, in scenarios where the number of labeled instances is constrained, a single model may lack the necessary discriminative power to assign pseudo-labels effectively to a large volume of unlabeled data [34]. To address this challenge our approach (Cross-Architecture Pseudo-Labeling in ActNetFormer) adopts a novel strategy of employing two models with distinct architectures and tasking them with generating pseudo-labels for each other. This approach is influenced by the understanding that different models exhibit distinct strengths and biases. While 3D CNNs excel in capturing spatial features and local dependencies within the temporal domain, transformers are more adept at handling long-range dependencies within the temporal domain. This variation in architectural characteristics leads to the generation of complementary semantic representations.

As shown in Fig. 2, we illustrate the ActNetFormer framework, which employs a cross-architecture setup. Specifically, we utilize the 3D-ResNet50 as the primary model  $Z(\cdot)$  and video transformer (ViT-S) as the auxiliary model  $A(\cdot)$ . Both models undergo supervised training using labeled data while simultaneously providing pseudo-labels for data unlabeled to their counterparts. This method encourages the two architectures to understand complementary representations, ultimately enhancing overall efficacy.

**Training on labeled data.** Training a model on labeled data involves a straightforward process. Given a set of labeled videos  $\{(x_i, y_i)\}_{i=1}^{B_l}$ , we define the supervised loss for both models as follows:

$$L_s^Z = \frac{1}{B_l} \sum_{i=1}^{B_l} \mathcal{H}(y_i, Z(G_n^Z(x_i))) \quad (2)$$

$$L_s^A = \frac{1}{B_l} \sum_{i=1}^{B_l} \mathcal{H}(y_i, A(G_n^A(x_i))) \quad (3)$$

where  $G_n(\cdot)$  denotes the conventional data augmentation method employed in [9, 29].

**Training on unlabeled data.** When presented with an unlabeled video  $u_i$ , the auxiliary model  $A(\cdot)$  generates predictions based on data that are weakly augmented  $u_i$  and produces category-wise probabilities denoted as  $q_i^A = A(G_w(u_i))$ . If the maximum probability among these probabilities,  $\max(q_i^A)$ , exceeds a predefined threshold  $\tau$ , it is considered a reliable prediction. In such cases, we utilize  $q_i^A$  to infer the pseudo ground truth label  $\hat{y}_i^A = \arg \max(q_i^A)$  for the strongly augmented  $u_i$ . This process allows the model  $Z(\cdot)$  to learn effectively.

$$L_u^Z = \frac{1}{B_u} \sum_{i=1}^{B_u} 1(\max(q_i^A) \geq \tau) \mathcal{H}(\hat{y}_i^A, Z(G_s(u_i))) \quad (4)$$

where,  $B_u$  represents the batch size, and  $\mathcal{H}(\cdot, \cdot)$  denotes the cross-entropy loss.

Similar to the auxiliary model, the primary model will also produce a prediction  $q_i^Z = Z(G_w(u_i))$ , which is then utilized to create a labeled pair  $(\hat{y}_i^Z, G_s(u_i))$  for the auxiliary model:

$$L_u^A = \frac{1}{B_u} \sum_{i=1}^{B_u} 1(\max(q_i^Z) \geq \tau) \mathcal{H}(\hat{y}_i^Z, A(G_s(u_i))) \quad (5)$$

**Contrastive learning.** The goal is to train the primary and auxiliary models using limited supervision initially, which can effectively analyze a vast collection of unlabeled videos to enhance activity understanding. Our cross-architecture pseudo-labeling approach already leverages two different architectures to capture different aspects of action representations as mentioned in Section 3.4. Contrastive learning is incorporated to encourage the models further to extract complementary features from the input data, leading to more comprehensive representations of actions. 3D CNN and a Video Transformer process the input video clip differently and produce a unique representation of the video content. In other words, the features extracted by each architecture capture different aspects of the video, such as spatial and temporal information. This diversity in representations can be advantageous as it allows the model to learn from multiple perspectives, potentially leading to a more comprehensive understanding of the action sequences in the videos. Therefore, cross-architecture contrastive learning is employed to discover the mutual information that coexists between both the representation encoding generated by the 3D CNN and the video transformer model. It is worth noting that our framework uses weakly augmented samples from each architecture for cross-architecture contrastive learning, inspired by [32].

Consider a mini-batch with  $B_u$  unlabeled videos. Here,  $m(u_i^Z)$  represents the video clip processed by the primary model, while  $m(u_i^A)$  represents the video clip processed by the auxiliary model. Therefore,  $m$  can be interpreted as the function that generates representations of the input videos through the respective models. These representations form the positive pair. For the rest of  $B_u - 1$  videos,  $m(u_i^Z)$  and  $m(u_k^A)$  form negative pairs, where the representation of the  $k$ -th video can come from either of the architecture

(i.e.,  $q \in \{Z, A\}$ ). Given that the negative pairs comprise various videos with distinct content, the representation of different videos within each architecture is pushed apart. This is facilitated by utilizing a contrastive loss ( $L_{ca}$ ) adapted from [5, 21], as outlined below.

$$L_{ca}(u_i^Z, u_i^A) = -\log \frac{h(m(u_i^Z), m(u_i^A))}{h(m(u_i^Z), m(u_i^A)) + \sum_{\substack{k=1 \\ q \in \{Z, A\}}}^B \mathbf{1}_{\{k \neq i\}} h(m(u_i^Z), m(u_k^q))} \quad (6)$$

where,  $h(u, v) = \exp\left(\frac{u^\top v}{\|u\|_2 \|v\|_2} / \tau\right)$  represents the exponential of the cosine similarity measure between vectors  $u$  and  $v$ , where  $\tau$  denotes the temperature hyperparameter. The final contrastive loss is calculated for all positive pairs,  $(u_i^Z, u_i^A)$ , where  $u_i^Z$  is the representation generated by the primary model and  $u_i^A$  is the representation generated by auxiliary model. The loss function is engineered to reduce the similarity, not just among different videos processed within individual architectures but also across both architectural models.

**Complete Training objective.** To encapsulate, merging supervised losses derived from labeled data with unsupervised losses derived from unlabeled data, we present the entire objective function as:

$$L = (L_s^Z + L_s^A) + \gamma \cdot (L_u^Z + L_u^A) + \beta \cdot L_{ca} \quad (7)$$

where,  $\gamma$  and  $\beta$  are weights of the cross-architecture loss and contrastive learning losses respectively.

## 4 Implementation

### 4.1 Auxiliary Model

As mentioned in Section 3.4, the auxiliary model should possess distinct learning capabilities compared to the primary model in order to offer complementary representations. Hence, we utilize VIT-S, which is the smaller version of the bigger transformer model (VIT-B). Comprehensive ablation studies (in the next section) show the superiority of VIT-S w.r.t. the transformer model (VIT-B) and the smaller 3D CNN model (3D-ResNet18). Unless otherwise specified, we utilize 3D-ResNet50 as the primary and VIT-S as the auxiliary models, respectively. More details of these models are included in Section S3 in the *Supplementary Material*.

### 4.2 Spatial data augmentations

We strictly adhere to the spatial data augmentations proposed in [9, 29] for training, denoted as  $G_n(\cdot)$ , on labeled data. For unlabeled data, random horizontal flipping, random scaling, and random cropping are employed as weak augmentations, denoted as  $G_w(\cdot)$ . The input size of the video is standardized to  $224 \times 224$  pixels to ensure consistency during augmentation and subsequent processing by the models. We utilize techniques such as AutoAugment [6] or Dropout [4] as strong augmentation,  $G_s(\cdot)$ .

### 4.3 Temporal data augmentations

Our ActNetFormer framework incorporates variations in frame rates for temporal data augmentations inspired by prior research in [21, 35]. While the primary model operates at a lower frame rate, the auxiliary model is provided with a higher one. This variation in frame rates allows for exploring different speeds in video representations. Despite the differences in playback speeds, the videos maintain the same semantics, maximizing the similarity between their representations. This approach offers complementary benefits by leveraging both slower and faster frame rates between the primary and auxiliary models. Consequently, this contributes to improving the overall performance of our ActNetFormer framework in action recognition. Additional spatial and temporal augmentations analysis are provided in Section S1.2 in *Supplementary Material*.

## 5 Experiments

We assess the effectiveness of the proposed ActNetFormer framework on three widely used datasets, *i.e.*, Kinetics-400 [14], HMDB-51 [15] and UCF-101 [23]. We employ two standard settings for semi-supervised action recognition, using 1% and 10% of the labeled data for the UCF-101 and Kinetics-400 datasets. However, for the HMDB-51 dataset, we use 40%, 50%, and 60% of the labeled data. Detailed ablation studies on the design choices of ActNetFormer are also conducted. Additionally, empirical analysis is provided in Section S2 in the *Supplementary Material* to validate the motivations behind ActNetFormer. It is crucial to emphasize that all experiments are conducted using a single modality (RGB only) and assessed on the corresponding validation sets unless stated otherwise.

### 5.1 Dataset

The Kinetics-400 dataset [14] comprises a vast collection of human action videos, encompassing around 245,000 training samples and 20,000 validation samples across 400 distinct action categories. Following established methodologies like MvPL [33] and CMPL [34], we adopt a labeling rate of 1% or 10%, selecting 6 or 60 labeled training videos per category. Additionally, the UCF-101 dataset [23] offers 13,320 video samples spread across 101 categories. We also sample 1 or 10 samples in each category as the labeled set following CMPL [34]. HMDB-51 [15] is a smaller dataset sourced from movie videos, featuring 51 human activity classes consisting of 6766 videos with high intra-class variance. We conduct experiments at three different labeling rates: 40%, 50%, and 60% based on LTG [32].

### 5.2 Baseline

For our primary model, we utilize the 3D-ResNet50 from [9]. We employ the ViT [8] extended with the video TimeSformer [2] as the auxiliary model in our ActNetFormer approach. While most hyperparameters remain consistent with the baseline, we utilize the divided space-time attention mechanism, as mentioned in TimeSformer [2]. However, only the big transformer model (ViT-B) is offered in TimeSformer, hence we adopt

the smaller transformer model (ViT-S) inspired by DeiT-S [27] with the dimensions of 384 and 6 heads. More details on the structure of primary and auxiliary models are included in Section S3 in the *Supplementary Material*.

### 5.3 Training and inference

During training, we utilize a stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 0.001. The confidence score threshold  $\tau$ , is set to 0.8. Parameters  $\gamma$  and  $\beta$  are both set to 2. Based on insights from the ablation study in Section 7.1, we employ a batch ratio of 1:5 for labeled to unlabeled data, ensuring a balanced and effective training process. A total of 250 training epochs are used. During testing, consistent with the inference method employed in MvPL [33] and CMPL [34], we uniformly sample five clips from each video and generate three distinct crops to achieve a resolution of  $224 \times 224$ , covering various spatial areas within the clips. The final prediction is obtained by averaging the softmax probabilities of these  $5 \times 3$  predictions. While both the primary and auxiliary models are optimized jointly during training, only the primary model is utilized for inference, thereby incurring no additional inference cost. It is noteworthy that our ActNetFormer approach does not rely on pre-training or pre-trained weights, setting it apart from other methods and underscoring its uniqueness in the field of action recognition in videos. We train our model entirely from scratch, further highlighting the robustness of our approach.

## 6 Results

**Table 1.** Comparison of results with SOTA approaches on UCF-101, Kinetics-400 and HMDB-51. The best-performing results are highlighted in red, while the second-best results are highlighted in blue. Methods utilizing pre-trained ImageNet weights are displayed in grey. "Params" indicates the number of parameters. "Input" shows the modality used during training, where "V" representing raw RGB video, "F" denoting optical flow, and "G" indicating temporal gradient. "Architecture" indicates the types of models used during training.

Method	Architecture	Input	Epoch	Params (M)	UCF-101			Kinetics-400			HMDB51		
					1%	10%	40%	1%	10%	40%	50%	60%	
FixMatch (NeurIPS 2020) [22]	3D-ResNet50	V	200	31.8	14.8	49.8	8.6	46.9	-	-	-	-	-
FixMatch (NeurIPS 2020) [22]	SlowFast-R50	V	200	60	16.1	55.1	10.1	49.4	-	-	-	-	-
TCL (CVPR 2021) [21]	TSM-ResNet-18	V	400	11.2	-	-	8.5	-	-	-	-	-	-
MvPL (ICCV 2022) [33]	3D-ResNet50	V+F+G	600	31.8	22.8	80.5	17.0	58.2	30.5	33.9	35.8	-	-
TACL (IEEE TCSVT 2022)[26]	3D-ResNet50	V	200	33.2	-	55.6	-	-	38.7	40.2	41.7	-	-
LTG (CVPR 2022) [32]	3D-ResNet18	V+G	180	67.1	-	62.4	9.8	43.8	46.5	48.4	49.7	-	-
CMPL (CVPR 2022) [34]	3D-ResNet50 + 3D-ResNet18	V	200	45.3	25.1	79.1	17.6	58.4	-	-	-	-	-
NCCL (IEEE TIP 2023) [31]	TSM-ResNet-18	V+G	400	23.1	21.6	-	12.2	43.8	-	-	-	-	-
DANet (Elsevier NN 2023) [10]	3D-ResNet18	V	600	31.8	-	64.6	-	-	-	-	-	-	-
<b>ActNetFormer (Ours)</b>	3D-ResNet50 + ViT-S	V	250	62.3	26.1	80.0	18.3	59.2	47.1	48.2	49.9	-	-
<b>ActNetFormer (Ours) with Contrastive learning</b>	3D-ResNet50 + ViT-S	V	250	62.3	27.6	80.6	19.1	59.8	47.9	49.1	51.1	-	-

We present the top-1 accuracy as our chosen evaluation metric in Table 1. ActNetFormer consistently performs better than various SOTA methods, including FixMatch,

TCL, MvPL, TAFL, CMPL, NCCL, DANet, and LTG, across all the three datasets and labeling rates. The inclusion of contrastive learning in our approach demonstrates an improved performance by a significant percentage, specifically at the 1% labeled data setting. We observe a percentage increase of approximately 4.60% for the UCF-101 and 4.37% for the Kinetics-400 dataset. This enhancement underscores the effectiveness of incorporating contrastive learning, resulting in more robust representations. ActNetFormer outperforms FixMatch by a large margin due to its novel cross-architecture strategy, which leverages the strengths of both 3D CNN and VIT models, whereas FixMatch relies solely on its own architecture for label generation, potentially limiting its adaptability. Our approach shares similarities with the CMPL approach. However, it surpasses CMPL in several vital aspects. Firstly, our approach incorporates video-level contrastive learning, which enables the model to learn more robust representations at the video level. This enhanced representation leads to better performance in action recognition. Additionally, our approach leverages a cross-architecture strategy, combining the strengths of both 3D CNN and VIT models. In contrast, CMPL leverages a cross-model strategy which utilizes the strength of 3D CNN alone. By integrating spatial feature extraction capabilities from CNNs with the attention mechanisms of transformers, our approach achieves a more comprehensive understanding of both spatial and temporal aspects of video data. Besides that, our approach achieves a performance of 80.0% in the 10% UCF-101 dataset, while incorporating contrastive learning boosts our performance to 80.6%, bringing it closer to the 80.5% achieved by MvPL. Notably, our approach relies solely on one modality, whereas MvPL exploits three modalities. Despite this discrepancy in input modalities, our approach demonstrates comparable performance, indicating its efficiency in leveraging single-modality information for video understanding tasks. This suggests that our approach may offer a more streamlined solution than MvPL, which relies on multiple modalities to achieve similar performance levels. Among the various approaches evaluated on the HMDB-51 dataset, the LTG method achieves the closest results to our ActNetFormer approach. Our ActNetFormer, without contrastive learning, performs slightly better at 40% and 60% labeled data, while LTG performs marginally better at 50% labeled data. However, with the addition of contrastive learning, ActNetFormer outperforms LTG across all three labeled data percentages. This demonstrates the substantial benefits of incorporating contrastive learning in our approach, leading to superior performance on the HMDB-51 dataset.

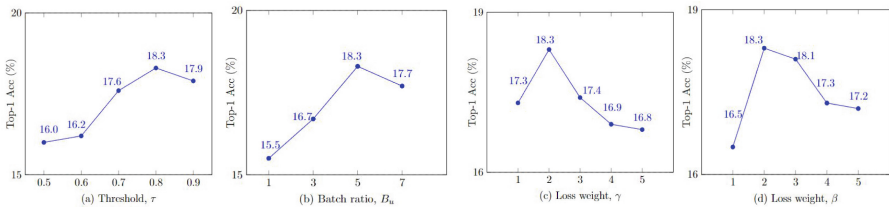
## 7 Ablation Studies

We thoroughly examine the proposed ActNetFormer method through several ablation studies. We present the experimental outcomes of various configurations of hyperparameters. We then analyze different combinations of the primary and auxiliary models. In all the ablation studies, it is crucial to highlight that experiments conducted with the UCF-101 dataset utilize 1% of the labeled data, while those conducted with the Kinetics-400 dataset also employ 1% of the labeled data.

## 7.1 Analysis of hyperparameters

Here, we investigate the impact of various hyperparameters. Experiments are conducted under the 1% setting of the Kinetics-400 dataset. Initially, we examine the influence of different threshold values of  $\tau$ . As illustrated in Fig. 3 (a), the results indicate that a threshold of ( $\tau = 0.8$ ) achieved the highest accuracy, suggesting that the quality of the threshold is crucial. Additionally, setting the threshold too high, as in the case of ( $\tau = 0.9$ ), may lead to sub-optimal performance, as evidenced by the lower accuracy compared to ( $\tau = 0.8$ ). When the threshold is set too high, there is a risk that only a limited number of unlabeled samples are selected for inclusion. This occurs because the threshold acts as a criterion for determining which samples are considered confidently predicted by the model and thus eligible for inclusion in the training process. Therefore, if the threshold is excessively high, fewer unlabeled samples may meet this criterion, leading to under-utilization of unlabeled data and potentially compromising model performance. Hence, we utilize 0.8 as the threshold for all the experiments in this study.

Next, we evaluate the impact of the ratio between labeled and unlabeled samples in a mini-batch on the final outcome. Specifically, we fix the number of labeled samples  $B_l$  at 1 and randomly sample  $B_u$  unlabeled samples to form a mini-batch, where  $B_u$  varies from  $\{1, 3, 5, 7\}$ . The outcomes are depicted in Fig. 3 (b), indicating that the model performs best when  $B_u = 5$ . Lastly, we explore the selection of the loss weights  $\gamma$  and  $\beta$ , as shown in Fig. 3 (c) and Fig. 3 (d) for the cross-architecture loss and contrastive learning loss, respectively. We find that the optimum value of  $\gamma$  and  $\beta$  are 2. Hence, we utilize  $\gamma = 2$  and  $\beta = 2$  for all the experiments.



**Fig. 3.** Analysis of different hyperparameters which includes Threshold  $\tau$ , Batch ratio  $B_u$ , Loss weight  $\gamma$ , Loss weight  $\beta$ .

## 7.2 Analysis of different combination of primary and auxiliary models used

“ResNet-B” explicitly denotes the 3D-ResNet50 model, while “ResNet-S” refers to the 3D-ResNet18 model. Correspondingly, “VIT-S” represents the smaller variant of the video transformer model, while “VIT-B” indicates the larger variant. Please keep these specific references in mind for clarity in our discussions. Before delving into the comparisons, it is important to note that we have critically analyzed why our approach (ResNet-B and VIT-S) outperforms other combinations. The comparison between



**Table 2.** Comparison of performance between primary and auxiliary models on UCF-101 (1%) and Kinetics-400 (1%) datasets.

Primary Model	Auxiliary Model	UCF-101 (1%)	Kinetics-400 (1%)
VIT-B	VIT-S	19.2	13.1
VIT-B	ResNet-B	20.9	13.9
VIT-B	ResNet-S	21.1	14.6
ResNet-B	VIT-B	23.7	16.9
ResNet-B	ResNet-S	25.1	17.6
<b>ResNet-B</b>	<b>VIT-S</b>	<b>26.1</b>	<b>18.3</b>

ResNet-B and VIT-S versus alternative combinations is illustrated in Table 2, and the analysis is detailed below.

The comparison between ResNet-B and VIT-S versus alternative combinations reveals detailed insights. ResNet-B and VIT-S, demonstrate the significance of cross-architecture approaches in video recognition tasks. Significant performance enhancements are achieved by leveraging ResNet-B’s spatial feature extraction and VIT-S’s temporal understanding. Additionally, VIT-S’s superiority as an auxiliary model highlights the effectiveness of smaller models, particularly in the temporal domain, due to its smaller parameter count and better suitability for scenarios with limited data. When VIT-B is employed as the primary model among the first three combinations, its best performance is achieved when paired with ResNet-S. This outcome validates our motivation for employing a cross-architecture strategy and demonstrates the efficacy of using smaller models as auxiliary components. The complementary nature and the efficacy in the temporal domain of smaller models enhance the overall performance. Overall, this analysis emphasizes the pivotal role of the cross-architecture approach and the utilization of smaller models in improving video recognition performance, aligning with the motivation of our study. Further ablations are provided in Section S1 of the *Supplementary Material*.

## 8 Conclusion

In conclusion, our proposed approach, ActNetFormer, combines cross-architecture pseudo-labeling with contrastive learning to offer a robust solution for semi-supervised video action recognition. By leveraging both labeled and unlabeled data, ActNetFormer effectively learns action representations by merging pseudo-labeling and contrastive learning techniques. This novel approach integrates 3D CNN and VIT to comprehensively capture spatial and temporal aspects of action representations. Additionally, cross-architecture contrastive learning is employed to explore mutual information between the encoding generated by 3D CNN and VIT. This strategy enhances the model’s ability to learn from diverse perspectives, resulting in superior performance. The success of ActNetFormer underscores the effectiveness of leveraging diverse architectures and semi-supervised learning paradigms to advance action recognition in real-world scenarios.

## References

1. Arnab, A., Deghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6936–6946 (2021)
2. Bertasius, G., Wang, H., Torresani, L.: Is space - time attention all you need for video understanding? In: ICML. vol. 3, p. 4 (2021)
3. Bilal, M., Maqsood, M., Yasmin, S., Hasan, N.U., Rho, S.: A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes. *J. Supercomput.* **78**(2), 2873–2908 (2022)
4. Bouthillier, X., Konda, K., Vincent, P., Memisevic, R.: Dropout as data augmentation (2016)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (13–18 Jul 2020), <https://proceedings.mlr.press/v119/chen20j.html>
6. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
7. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. *Comput. Vis. Image Underst.* **219**, 103406 (2022)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
9. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)
10. Gao, G., Liu, Z., Zhang, G., Li, J., Qin, A.K.: Danet: Semi-supervised differentiated auxiliaries guided network for video action recognition. *Neural Netw.* **158**, 121–131 (2023)
11. Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T., Ding, R.: Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. Proceedings of the AAAI Conference on Artificial Intelligence **36**(1), 762–770 (2022). <https://doi.org/10.1609/aaai.v36i1.19957>. <https://ojs.aaai.org/index.php/AAAI/article/view/19957>
12. Hu, Z., Yang, Z., Hu, X., Nevatia, R.: Simple: Similar pseudo label exploitation for semi-supervised classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15099–15107 (June 2021)
13. Jing, Y., Wang, F.: Tp-vit: A two-pathway vision transformer for video action recognition. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2185–2189. IEEE (2022)
14. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint [arXiv:1705.06950](https://arxiv.org/abs/1705.06950) (2017)
15. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International conference on computer vision. pp. 2556–2563. IEEE (2011)
16. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: A framework and review. *Ieee Access* **8**, 193907–193934 (2020)
17. Pareek, P., Thakkar, A.: A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artif. Intell. Rev.* **54**, 2259–2322 (2021)
18. Rao, H., Xu, S., Hu, X., Cheng, J., Hu, B.: Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. *Inf. Sci.* **569**, 90–109 (2021)

19. Shah, A., Sra, S., Chellappa, R., Cherian, A.: Max-margin contrastive learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8220–8230 (2022)
20. Shen, H., Yan, Y., Xu, S., Ballas, N., Chen, W.: Evaluation of semi-supervised learning method on action recognition. *Multimedia Tools and Applications* **74**, 523–542 (2015)
21. Singh, A., Chakaborty, O., Varshney, A., Panda, R., Feris, R., Senko, K., Das, A.: Semi-supervised action recognition with temporal contrastive learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10389–10399 (2021)
22. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural. Inf. Process. Syst.* **33**, 596–608 (2020)
23. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint [arXiv:1212.0402](https://arxiv.org/abs/1212.0402) (2012)
24. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? *Adv. Neural. Inf. Process. Syst.* **33**, 6837–6839 (2020)
25. Tian, Y.: Understanding deep contrastive learning via coordinate-wise optimization. *Adv. Neural. Inf. Process. Syst.* **35**, 19511–19522 (2022)
26. Tong, A., Tang, C., Weng, W.: Semi-supervised action recognition from temporal augmentation using curriculum learning. *IEEE Trans. Circuits Syst. Video Technol.* **33**(3), 1301–1312 (2023). <https://doi.org/10.1109/TCSVT.2022.3310271>
27. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (18–24 Jul 2021), <https://proceedings.mlr.press/v139/touvron21a.html>
28. Varshney, N., Bakariya, B.: Deep convolutional neural model for human activities recognition in a sequence of video by combining multiple cnn streams. *Multimedia Tools and Applications* **81**(29), 42117–42129 (2022)
29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
30. Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X.: Semi-supervised semantic segmentation using unreliable pseudo-labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4248–4257 (June 2022)
31. Wu, J., Sun, W., Gan, T., Ding, N., Jiang, F., Shen, J., Nie, L.: Neighbor-guided consistent and contrastive learning for semi-supervised action recognition. *IEEE Transactions on Image Processing* (2023)
32. Xiao, J., Jing, L., Zhang, L., He, J., She, Q., Zho, Z., Yuile, A., Li, Y.: Learning from temporal gradient for semi-supervised action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3252–3262 (2022)
33. Xiong, B., Fan, H., Grauman, K., Feichtenhofer, C.: Multiview pseudo-labeling for semi-supervised learning from video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7209–7219 (October 2021)
34. Xu, Y., Wei, F., Sun, X., Yang, C., Shen, Y., Dai, B., Zhou, B., Lin, S.: Cross-model pseudo-labeling for semi-supervised action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2959–2968 (2022)
35. Yang, C., Xu, Y., Dai, B., Zho, B.: Video representation learning with visual temporal consistency. arXiv preprint [arXiv:2006.15599](https://arxiv.org/abs/2006.15599) (2020)
36. Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., Shinozaki, T.: Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Adv. Neural. Inf. Process. Syst.* **34**, 18408–18419 (2021)

37. Zhang, T., Liu, S., Xu, C., Lu, H.: Boosted multi-class semi-supervised learning for human action recognition. *Pattern Recogn.* **44**(10–11), 2334–2342 (2011)
38. Zheng, S., Chen, S., Jin, Q.: Few-shot action recognition with hierarchical matching and contrastive learning. In: *European Conference on Computer Vision*. pp. 297–313. Springer (2022)
39. Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., Zhang, Z., Tighe, J., Manmatha, R., Li, M.: A comprehensive study of deep video action recognition. arXiv preprint [arXiv:2012.06567](https://arxiv.org/abs/2012.06567) (2020)



# RSTAN: Residual Spatio-Temporal Attention Network for End-to-End Human Fall Detection

Yaru Jiang<sup>1</sup>, Shujing Lyu<sup>1</sup>✉, Hongjian Zhan<sup>1</sup>, and Yue Lu<sup>1</sup>

Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai, China  
51255904028@stu.ecnu.edu.cn, {sjlv, ylu}@cs.ecnu.edu.cn,  
ecnuhjzhan@foxmail.com

**Abstract.** The occurrence of human fall is a significant threat to human health, especially among the elderly. Unlike standard action recognition, falls manifest a combination of static and dynamic attributes. They are highly sensitive to spatio-temporal motion, marked by sudden and transient occurrences. This paper proposes a novel spatio-temporal convolutional method for end-to-end human fall detection, named Residual Spatio-Temporal Attention Network (RSTAN). The network integrates a Spatial Channel Attention (SCA) module within the convolutional layers of the Residual 3D convolution to enhance feature refinement. selectively accentuates spatial and channel dimensions. In addition, to capture both the extensive spatio-temporal features and the short-range spatio-temporal characteristics of human falls, effectively distinguishing them from daily activities, we propose a Multi-interval Difference Aggregation (MDA) method. The MDA utilizes multiple time interval frame differences to extract motion features. Our proposed method's superior performance is demonstrated through experiments on three publicly available fall detection datasets. Specifically, achieving 100% accuracy on the UR Fall Detection dataset.

**Keywords:** Human fall detection · Residual 3D convolution · Spatial channel attention · Multi-interval difference aggregation

## 1 Introduction

Fall detection plays a vital role in safeguarding individuals, particularly the elderly. The World Health Organization (WHO) acknowledges this critical issue, reporting approximately 37.3 million falls annually that necessitate medical care. Falls are the second leading cause of unintentional injury-related deaths globally[30]. International guidelines for fall prevention and management in older adults emphasize the potential benefits of e-health technologies, including wearables, virtual reality applications, and environmental monitoring systems[16].

Fall detection systems for humans are primarily classified into three categories: wearable sensor-based systems[19–21], ambient device-based systems[5,

6,11,17], and camera device-based systems[13]. Comparing wearable devices, ambient device-based systems primarily collect wifi and MEMS information to detect fall events, which do not restrict human activity. However, installing these sensors throughout the entire area can be laborious. In contrast to wearable devices and ambient device-based systems, camera device-based systems utilize existing cameras or surveillance cameras to capture video data and offer non-invasiveness and low cost for the elderly. Additionally, visual detection systems can provide more intuitive and comprehensive information, with higher accuracy and reliability.

In vision-based research, fall detection often rely on depth images acquired through RGB-D sensors, as these sensors provide rich information about human motion patterns via depth maps. However, recent advancements in deep learning-based algorithms have enabled many studies to achieve significant performance in Human fall detection using conventional RGB cameras. However, deep learning-based methods often perform fall detection in two stages. The first stage is based on pos-Net or Yolo-pos[25] to extract human skeleton features, and the second stage is based on LSTM[12] or CNN classification[31,33,34]. Alternatively, the process may commence with employing Yolo[25] to detect the human box, followed by LSTM to establish continuity relationships between frames[4,7,15,29], facilitating the identification of falling frames. Moreover, motion features can be extracted through frame difference or optical flow techniques, and subsequently convolved to classify falling frames. Compared with complex integration methods, accurate and sensitive end-to-end fall detection is needed to be more flexible for real-world deployment in fall detection systems.

Fall events cannot be adequately distinguished solely based on the magnitude of motion, as complex foreground information from various behaviors such as running and lying complicates the analysis. Hence, there is a need for the model to discern human spatial velocity, enabling the differentiation of fall events from other daily activities characterized by intricate spatial dynamics. We introduce the Residual Spatio-Temporal Attention Network (RSTAN) design based on the Spatial Channel Attention (SCA) mechanism and a residual 3D convolution, named R(2+1)D[23]. In this network, SCA offers significant advantages by enhancing feature representation and concurrently mitigating the detrimental impact of noise and irrelevant spatial features. The incorporation of SCA enhances the overall efficiency and effectiveness of video-based R(2+1)D convolutional networks for fall detection applications. Moreover, in contrast to the slow extraction process of optical flow motion features, our method, Multi-interval Difference Aggregation (MDA), is designed to facilitate the rapid acquisition of rich spatial motion characteristics.

**In summary, the key contributions of this paper are as follows:**

1. We replace the complex integrated model with an end-to-end spatio-temporal convolutional network for human fall detection while improving fall detection accuracy.

2. We design a RSTAN based on SCA to dynamically adjust the network's focus on different channels, which enhances the performance of the baseline model comprehensively.
3. We design the MDA method to capture subtle spatial velocity features of human falls, distinguishing them from daily activities.
4. We complete extensive experiments on three publicly available fall detection datasets to demonstrate the superior performance of our method.

## 2 Related works

Fall detection is integral to intelligent monitoring and home security systems. Some studies detect falls by extracting human skeletal pose and classifying them. Yadav et al. [33] preprocessed the pose coordinates and fed them into specially designed convolutional neural networks (CNNs) along with gated recurrent units (GRUs) in a sliding window manner, enabling the models to capture the spatiotemporal patterns within the raw data. Recent research, Noor et al. [18] proposed a lightweight skeleton-based 3D-CNN fall recognition network that demonstrates significant improvements in accuracy and processing time. This reflects the importance of 3D-CNN in fall detection. However, these methods are all based on two-stage approaches for fall detection. In contrast, action recognition offers an end-to-end fall detection method.

In action recognition, numerous studies have developed innovative spatio-temporal network architectures designed to enhance the learning of temporal features. Tran et al. [22] illustrated the advantages of 3D CNNs over 2D CNNs for extracting temporal features by integrating the time dimension directly into the CNN framework. A subsequent advancement, the Inflated 3D Convolution (I3D) [2, 27], extended traditional 2D convolution kernels into 3D space to effectively capture spatiotemporal features. Similarly, the R(2+1)D [23] method sought to decompose 3D convolutions into a series of spatial and temporal convolutions, aiming to reduce parameter count while maintaining performance.

Based on video feature extraction in action recognition, some researchers have experimented with end-to-end fall detection. Wang et al. [26] offers a forward-thinking, end-to-end method for video feature extraction and classification in fall detection. Their approach defines a fall merely as a deviation of the body's center of gravity from the vertical line, along with an inability to maintain balance. However, this definition could oversimplify real-world complexities and may lack the universality needed for fall detection across diverse scenarios.

While fall detection methodologies share similarities with those used in action recognition, the unique nature of falls demands special consideration. Falls manifest as a blend of static and dynamic characteristics, and are susceptible to motion characteristics, characterized by sudden and transient occurrences. Therefore, an end-to-end fall detection network, which is based on video understanding for action recognition and is sensitive to motion information, is necessitated.

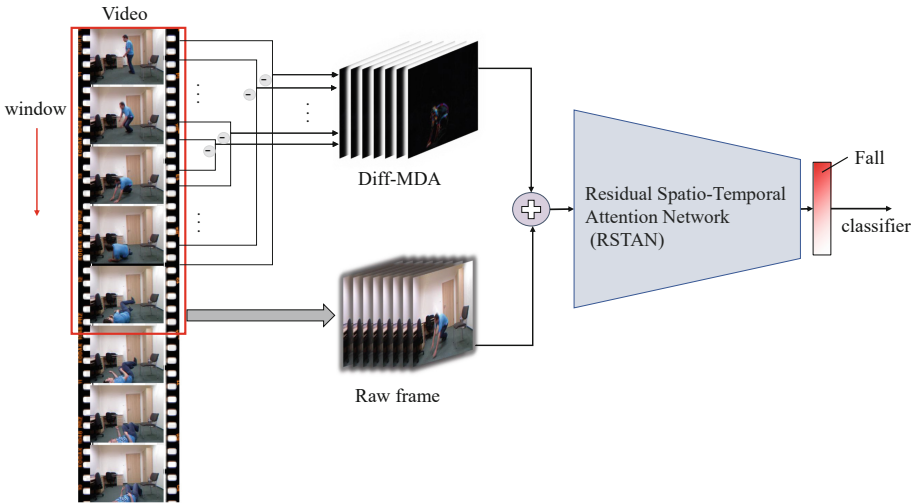
### 3 Method

In this study, we present a novel human fall detection architecture named Residual Spatio-Temporal Attention Network (RSTAN) and Multi-interval Difference Aggregation (MDA). In the subsequent section, we provide a comprehensive introduction to the framework of the proposed method.

#### 3.1 Overview

Because the fall process has a great correlation with the motion characteristics of humans, compared with the slow extraction of optical flow motion features, we extract frame difference features to obtain motion features quickly. Furthermore, the accuracy of motion features is inherently affected by the frame difference interval. For slow-moving objects, longer interval frame differences are suitable, while shorter intervals are preferable for fast-moving objects. Therefore, we design an MDA module, to capture fall actions at different speeds. This module extracts rich motion information by calculating the frame differences over various time intervals.

After that, we input the multi-interval frame difference image and the original image into the RSTAN and then get the classification result of whether the human is falling or not. We give an illustration of this architecture (see Fig.1).



**Fig. 1. The proposed human fall detection architecture.** Where " $\ominus$ " denotes the frame difference operation across multiple frame intervals, motion features are extracted within various temporal ranges in this way for the result of "Diff-MDA"; " $\oplus$ " symbolizes the concatenation operation.

Given that the typical human fall action occurs within 1 to 2 seconds, and considering a video frame rate of 25 FPS, we opt to utilize a sliding window



approach with a length of 16 frames, as the input clip length for the network. Additionally, we set the step size to 1-4, ensuring the accuracy of fall detection.

### 3.2 The Design of the Multi-Interval Differential Aggregation

It has been observed that the primary distinguishing feature between falls and routine activities (such as lying down, sitting, or sleeping) lies in the rapid vertical downward movement of the individual’s posture. Consequently, to capture the motion attributes associated with falls, we expedite the extraction of motion features through frame difference. However, it is noteworthy that when falls occur swiftly, smaller frame difference intervals are necessary to effectively detect this action. Conversely, when falls happen gradually, larger frame difference intervals are required. To address this variability, we integrate information from multiple time frame differences, enabling the detection of fall actions across different speeds.

Firstly, the input video sequence is segmented into a series of image frames. For each pair of consecutive frames (such as  $\mathbf{F}_t$  and  $\mathbf{F}_{t+1}$ ), the pixel-level differences between them are calculated to form a frame of differential image.

$$\mathbf{V}(t) = \mathbf{V}(:, :, \mathbf{F}_t, \mathbf{F}_{t+1}, :, s), t \in (1, Clip) \quad (1)$$

To capture movements at different speeds, the MDA module does not merely calculate the differences between adjacent frames but computes the differences between frames at multiple intervals. The interval  $i$  is continuously increased to realize the frame difference images with  $(1, 3, 5, \dots, clip - 1)$  intervals. Finally, they are aggregated into the  $\frac{clip}{2} * 3 * 112 * 112$  shape data, and input into the R(2+1)D network based on spatial channel attention.

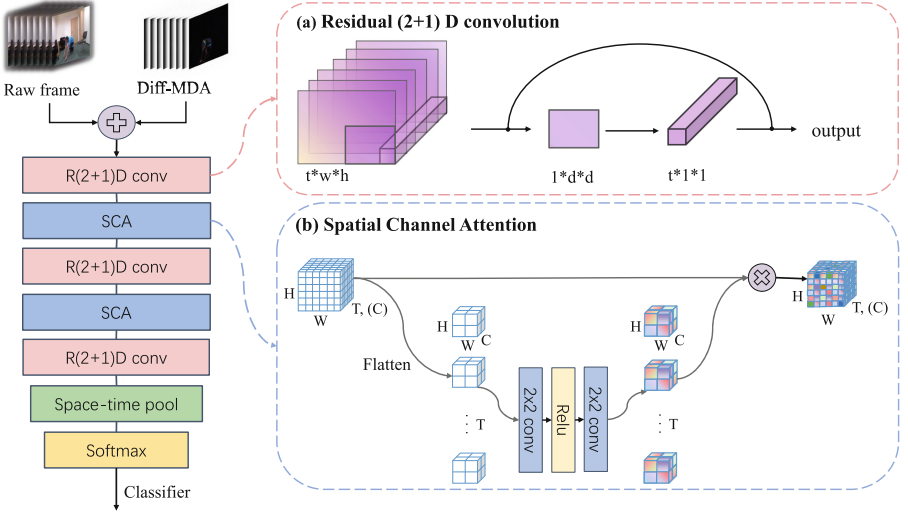
$$\mathbf{D}_{2i} = |\mathbf{V}(clip - 1 - i) - \mathbf{V}(i)|, i \in (1, Clip/2) \quad (2)$$

### 3.3 The Framework of Residual Spatio-Temporal Attention Network

The RSTAN is incorporated by a Spatial Channel Attention (SCA) module between the convolutional layers of the Residual 3D convolution. The framework of RSTAN is shown to illustrate (see Fig.2).

Different from the typical Residual 3D convolution, We choose R(2+1)D[23] as our baseline. R(2+1)D convolution separates the spatial and temporal dimensions, which reduces the number of model parameters while maintaining an efficient feature extraction capability. In this network, R (2+1) D is divided into  $1 \times d \times d$  2D spatial convolution and  $t \times 1 \times 1$  temporal convolution based on R3D, which has higher modeling flexibility and fewer parameters.

In order to dynamically adjust the network’s focus on different channels, we introduce a spatial channel attention mechanism to thereby enhance the recognition capability for falling actions. Upon obtaining the output from the R(2+1)D convolution, a flattening operation is performed on the output. In



**Fig. 2. The framework of RSTAN.** (a) Focuses on the residual (2+1) D convolution, R(2+1)D realizes three-dimensional spatio-temporal convolution by spatial two-dimensional convolution and temporal one-dimensional convolution. (b) Visual illustration of SCA, where we use different colors to represent the attention weights, "⊗" denotes matrix multiplication.

contrast to conventional average pooling, the flatten operation does not lose any information but instead flattens all features into a one-dimensional vector, effectively preserving all the information from the original feature map. This flattened vector is denoted as  $\mathbf{F} \in \mathbb{R}^{N \times T \times H \times W}$ . Subsequently, a shared Multi-Layer Perceptron (MLP) is employed, which consists of two  $2 \times 2$  convolutional layers with a ReLU activation function in between. The MLP transforms  $\mathbf{F}$  into a set of feature weights  $\mathbf{W} \in \mathbb{R}^{N \times T \times H \times W}$ . This spatial channel attention is illustrated above (see Fig.2(b)), mathematically expressed as:

$$\mathbf{W} = \text{Conv}_{2 \times 2}(\text{ReLU}(\text{Conv}_{2 \times 2}(\mathbf{F}))) \quad (3)$$

Finally, the feature weights  $\mathbf{W}$  are used to reweight the original input to the attention mechanism. This is achieved by summing over the product of the feature weights and the corresponding input features, for all spatial positions in the input. The final output of the SCA,  $\mathcal{V}_{SC}(\mathbf{X})$ , is given by:

$$\mathcal{V}_{SC}(\mathbf{X}) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \mathbf{W}_{i,j} \cdot \mathbf{X}_{i,j}^t \quad (4)$$

Where  $\mathcal{V}_{SC}(\cdot)$  denotes the function operation applied to the spatial channel,  $\mathbf{X}_{i,j}^t$  is the input feature at spatial position  $(i, j)$ , and  $\mathbf{W}_{i,j}$  is the corresponding feature weight, while  $K$  denotes the size of the 2-D convolution kernel.

In the proposed framework, the SCA is integrated into the Residual 3D Convolutional Network (R3D). Specifically, the SCA is applied on the output of the original R3D network, and then the result is fed back into the R3D network. This process can be mathematically expressed as:

$$R3D(X) = R3D_{\text{old}}(\mathcal{V}_{SC}(R3D_{\text{old}}(X))) \quad (5)$$

where  $R3D_{\text{old}}(X)$  denotes the output of the original R3D network processing the input  $X$ ,  $\mathcal{V}_{SC}(X)$  represents the function of the SCA module, and  $R3D(X)$  denotes the output of the R3D network with the integrated SCA module processing the input  $X$ .

### 3.4 Loss Function.

Given the substantial volume of daily activity videos within the fall dataset compared to the limited quantity of fall-related data, addressing the class imbalance issue is imperative. To mitigate this challenge, we employ the weighted cross-entropy loss function. We set the inverse ratio of the fall and daily action category proportions for the weights assigned to the respective categories to achieve the best model performance. The loss function as follows:

$$loss = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m W_j y_{ij} \log \hat{y}_{ij} \quad (6)$$

Here,  $y_{ij}$  is the true label of the  $j$ -th class of the  $i$ -th sample,  $\hat{y}_{ij}$  is the predicted value of the  $j$ -th class of the  $i$ -th sample,  $W_j$  represents the  $j$ -th class weight,  $n$  is the number of samples, and  $m$  is the number of classes.

## 4 Experiments

### 4.1 Dataset

- (1) The UR Fall Detection Dataset (URFD) was developed by the Computational Modeling Discipline Centre at the University of Rzeszow [14]. This dataset contains 70 video clips in total, with 30 showing falls and 40 illustrating non-fall activities like walking, sitting, squatting, and leaning. The individuals in the videos display a range of fall behaviors simultaneously, such as leaning backward, tilting, and suddenly collapsing to the ground. All the recorded activities, including both falls and daily actions, are captured in RGB images with a resolution of  $640 \times 480$ .
- (2) The Le2i dataset, developed by Charfi et al. [3], comprises 191 video sequences featuring multiple actors and four distinct stages, unlike other datasets. These videos include variations in lighting conditions and present typical challenges such as occlusions and cluttered or textured backgrounds. The actors perform a range of normal daily activities alongside fall events. The videos are recorded at a frame rate of 25 FPS with a resolution of  $320 \times 240$  pixels.

- (3) The Multiple Cameras Fall Dataset (MCFD), developed by Auvinet et al. [1], consists of 192 video sequences across 24 scenes, including 22 fall scenarios and two scenes depicting daily activities. These videos were captured using eight calibrated cameras, each with a resolution of  $720 \times 480$  pixels. This dataset is unique in offering a wide variety of perspectives on behavior and motion, encompassing not only typical fall-related actions but also activities like moving boxes, running, and cleaning rooms.

**Dataset process:** We process the dataset to train the fall detection model. Initially, We segment the video sequences into three categories: before the fall, during the fall, and after the fall. The video sequences before and after the fall are classified as daily activity, while the sequences during the fall are classified as falls. Due to the excessive length of many daily activity sequences, and considering that falls usually last only 1 to 2 seconds, we divide the excessively long videos into sequential slices of 32 frames each. These segmented data are then divided into training, validation, and testing sets for experimentation. The detailed post-processing datasets are presented in Table 1.

**Table 1.** Detailed descriptions of the number of video categories after dataset processing in the three publicly available fall detection datasets.

Datasets	URFD			Le2i			MCFD		
	train	val	test	train	val	test	train	val	test
ADL video	197	57	87	501	168	167	5287	1180	1940
Fall video	19	5	6	36	6	11	175	43	46
Total video	216	62	93	537	174	178	5462	1223	1986

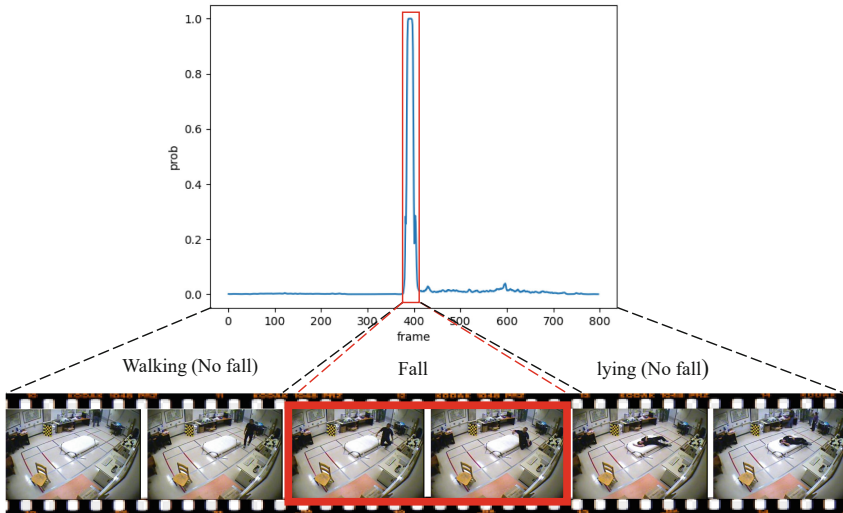
## 4.2 Implementation Details

We leverage the PyTorch framework with an NVIDIA RTX3090Ti GPU to implement our experiment. For each input video, we extract continuous  $T$  frames to form a clip, which is then input into our model. Each frame is resized to  $112 \times 112$ . The input fed to the model is of the size  $B \times T \times 3 \times 112 \times 112$ . The train batch size  $B$  is set to 24 and clip  $T$  sets 16. We adopt Adam as an optimizer and the learning rate is set to 0.01. Since the input class is imbalanced, we use the weighted cross-entropy loss as the loss function and set the inverse ratio of the fall and no fall category proportions for the weights assigned to the respective categories to achieve a better model performance.

## 4.3 Experimental Visualization

To test the performance of our model, we randomly select the long video data corresponding to the test data from the dataset, and then the model sends it

to the model with a sliding window of 16 frames in length to obtain the fall classification score prob of the current window and the classification score of no fall. The final test result is determined by the label corresponding to the highest probability (prob) score. Above the video sequences are the visualized detection prob scores from testing (see Fig. 3).



**Fig. 3. Visualization of fall detection results.** The x-axis represents the number of frames, while the y-axis represents the video classification results, specifically the fall probability score within the sliding window, denoted by "prob". We mark the falling video sequence with red boxes.

It is the classification score of the fall label obtained from a long video of nearly 800 frames (for the binary classification model, the prob score of no fall is  $1 - fall$  prob). Around the 400th frame, the person falls, and at this point, the probability score of fall increases rapidly. Meanwhile, in other video sequences of daily activities where no falls occurred, prob scores are very low. It can be seen that the model is accurate in fall localization and very sensitive to fall detection.

#### 4.4 Performance Comparison with Existing Approaches

We compared our method with previous state-of-the-art fall detection approaches using the UR Fall Detection Dataset, Multiple Cameras Fall Dataset, and Le2i Fall Detection Dataset. The methods selected for comparison include the leading fall detection techniques commonly applied to these datasets. To assess the accuracy of our method in classifying fall videos, we employed accuracy, recall, precision, and F1-score as the evaluation metrics.

It is obtained by experimental results in Table 2, our method achieves optimal accuracy and sensitivity of 100% on the URFD Fall Detection Dataset. It

**Table 2.** Comparison of the proposed method with existing methods on three public fall datasets.

Dataset	Methods	Accuracy	Recall	Precision	F1 score
URFD	Pose+GAN [31]	0.885	0.821	0.934	-
	YOLOV3+LSTM[8]	-	0.914	0.948	0.931
	HOP+MBH[24]	-	0.975	0.969	0.971
	CNN+LSTM[32]	-	0.967	0.979	0.973
	YOLOK+3DCNN[9]	<u>0.9966</u>	<u>0.9949</u>	<u>0.9992</u>	-
	Ours	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>
Le2i	Pose+GAN[31]	0.916	0.80	0.926	-
	CNN+LSTM[32]	<b>0.984</b>	0.93	<b>0.993</b>	<b>0.984</b>
	RNN [10]	-	<b>0.99</b>	0.97	-
	Ours	<u>0.9831</u>	<u>0.98</u>	<u>0.98</u>	<u>0.98</u>
MCFD	RNN[10]	-	<u>0.98</u>	0.96	-
	YOLOv3+LSTM[8]	-	0.916	0.935	-
	YOLOK+3DCNN[9]	<u>0.9822</u>	<b>0.9862</b>	<b>0.9777</b>	-
	Ours	<b>0.9848</b>	<u>0.98</u>	<u>0.97</u>	0.97

also shows excellent performance on the other two public datasets and proves the excellent ability of our proposed network in capturing spatio-temporal motion information. A series of comparative experiments have demonstrated the exceptional ability of our proposed RSTAN to capture spatio-temporal motion information.

It’s worth noting that our method achieved the second-highest performance on the Le2i dataset. Through our visualization experiments, we discerned substantial fluctuations in lighting conditions within the Le2i dataset. This pronounced variability in scene illumination appears to be a significant factor contributing to the misclassifications made by our model. The primary cause of this issue lies in our network’s heightened sensitivity to motion features, leading it to erroneously interpret changes in a person’s shadow as a fall event. To address this challenge, future research should concentrate on fortifying the network’s capacity for robust visual target comprehension and reduce the model’s vulnerability to such environmental alterations, thereby improving its overall performance and reliability.

#### 4.5 Ablation Study

To demonstrate the efficacy of our methodology, we will carry out a comprehensive set of ablation experiments using publicly available datasets. This rigorous analysis aims to thoroughly examine and validate various components of our proposed approach. Specifically, we will examine various components within our architecture, such as the RSTAN with spatial channel attention module or with-

out spatial channel attention. We also experimented with MDA and other feature extraction methods for comparison. Following this analysis, we aim to identify the optimal configuration of the proposed method.

**Impact of Multi-interval Difference Aggregation.** In the fall detection experiments, we utilize the original RGB frames, optical flow, frame difference, and a combination of RGB and frame difference frames as inputs to train our network. During validation and testing, the accuracy is calculated after the same video processing. In the experimental design involving the combination of RGB and frame difference frames, we set the frame difference intervals to 2, 4, and 8, along with our proposed MDA method. Where optical flow is generated using the denseflow[28] method. We conducted our experiments on the URFD, Le2i, and MCFD fall detection datasets, considering MDA and other input modalities. The data presented in the table represents the accuracy rate.

**Table 3.** Ablation study regarding MDA and other input modalities on the URFD/Le2i/MCFD fall detection dataset.

Input Mode	RGB	Optical Flow	Frame Diff	RGB+Diff4	RGB+Diff8	RGB+MDA
URFD	0.9876	0.9436	0.9876	<u>1.0</u>	<u>1.0</u>	<b>1.0</b>
Le2i	<u>0.9551</u>	0.9494	0.9438	<u>0.9551</u>	0.9494	<b>0.9831</b>
MCFD	0.9517	0.8319	0.9613	<u>0.9773</u>	0.9768	<b>0.9848</b>

As evidenced by our experimental results from Table 3, it is observed that the approach of employing “RGB+diff” as the feature input modalities always yield superior accuracy. This combination leverages the strengths of both static and dynamic information, enabling the model to better identify the rapid movements characteristic of fall events. Our multi-scale frame difference aggregation approach further refines this by capturing motion information at various scales, which is crucial for accurately detecting falls.

In contrast, relying solely on RGB static features or diff motion features is insufficient. RGB features alone lack the necessary temporal information to capture motion dynamics, while diff features alone may miss critical spatial details. Additionally, while optical flow features theoretically provide comprehensive motion information, the increased computational complexity does not translate into improved performance, making it an inefficient choice for this task.

Overall, the experimental results underscore the importance of combining multiple types of features and employing multi-scale aggregation to achieve robust fall detection.

**Impact of Spatial Channel Attention Module.** We also consider the Neural Network and compare different situations: RSTAN with SCA or without. When other parameters are set the same, across all evaluations conducted on

fall detection datasets, the incorporation of SCA consistently yields superior performance compared to scenarios where it is not utilized.

**Table 4.** The results of ablation study regarding SCA of the proposed method on the URFD/Le2i/MCFD fall detection dataset.

Dataset	Measures	Accuracy	Recall	Precision	F1 score
URFD	w/o SCA	0.9892	0.99	0.99	0.99
	w/ SCA	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>
Le2i	w/o SCA	0.9212	0.92	0.93	0.92
	w/ SCA	<b>0.9831</b>	<b>0.98</b>	<b>0.98</b>	<b>0.98</b>
MCFD	w/o SCA	0.9507	0.95	0.97	0.96
	w/ SCA	<b>0.9848</b>	<b>0.98</b>	<b>0.98</b>	<b>0.98</b>

According to the test results in Table 4, we can analyze the SCA offering significant benefits in enhancing feature representation and discriminative capability. By selectively emphasizing informative spatial regions while suppressing irrelevant ones, SCA effectively enhances the spatial discriminative power within the convolutional feature maps. This attention mechanism enables the network to focus on salient spatial information, thereby facilitating more robust and accurate feature extraction in the spatiotemporal domain.

## 5 Conclusion And Future Work

In this paper, we introduce a novel end-to-end fall detection method that leverages the Multi-Interval Difference Aggregation (MDA) and Residual Spatio-Temporal Attention Network (RSTAN), integrating residual 3D convolution with Spatial-Channel Attention (SCA) mechanisms. Experimental results show that our approach achieves outstanding performance on three public fall detection datasets. Additionally, ablation studies confirm the effectiveness of both the MDA and SCA modules.

In our future work, we aim to optimize the model structure to enhance detection speed and reduce computational resource consumption. Additionally, we will pursue the development of more lightweight models to enable hardware deployment with lower resource requirements and costs.

**Acknowledgments.** This work was supported by the Science and Technology Commission of Shanghai Municipality, under Grant 22DZ2229004, and Shanghai Trusted Industry Internet Software Collaborative Innovation Center.



## References

1. Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Multiple cameras fall data set
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4733. IEEE, Honolulu, HI (Jul 2017). <https://doi.org/10.1109/CVPR.2017.502>
3. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Optimised spatio-temporal descriptors for real-time fall detection: comparison of svm and adaboost based classification. *Journal of Electronic Imaging (JEI)* **22**(4), 17 (2013)
4. Chen, G., Sun, S., Sun, Y., Chen, H., Zhang, W., Su, X.: Cagn: High-order coordinated attention module for improving fall detection models. In: IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society. pp. 1–<https://doi.org/10.1109/IECON51785.2023.10311741>
5. Chen, S., Yang, W., Xu, Y., Geng, Y., Xin, B., Huang, L.: Afall: Wi-fi-based device-free fall detection system using spatial angle of arrival (8), 4471–4484,<https://doi.org/10.1109/TMC.2022.3157666>
6. Chu, Y., Cumanan, K., Sankarpani, S.K., Smith, S., Dobre, O.A.: Deep learning-based fall detection using wifi channel state information. *IEEE Access* **11**, 83763–83780 (2023).<https://doi.org/10.1109/ACCESS.2023.3300726>
7. Chutimawattanakul, P., Samanpiboon, P.: Fall detection for the elderly using yolov4 and lstm. In: 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). pp. 1–5.<https://doi.org/10.1109/ECTI-CON54298.2022.9795534>
8. Feng, Q., Gao, C., Wang, L., Zhao, Y., Song, T., Li, Q.: Spatio-temporal fall event detection in complex scenes using attention guided LSTM pp. 242–249.<https://doi.org/10.1016/j.patrec.2018.08.031>
9. Gomes, M.E.N., Macêdo, D., Zanchettin, C., de Mattos-Neto, P.S.G., Oliveira, A.: Multi-human fall detection and localization in videos. *Comput. Vis. Image Underst.* **220**, 103442 (2022). <https://doi.org/10.1016/j.cviu.2022.103442>
10. Hasan, M.M., Islam, M.S., Abdullah, S.: Robust pose-based human fall detection using recurrent neural network. In: 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON). pp. 48–51.<https://doi.org/10.1109/RAAICON48939.2019.23>
11. He, C., Liu, S., Zhong, G., Wu, H., Cheng, L., Yan, G., Wen, Y.: A noncontact fall detection method for bedside application with a mems infrared sensor and a radar sensor (14), 12577–12589.<https://doi.org/10.1109/JIOT.2023.3251980>
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
13. Kong, Y., Huang, J., Huang, S., Wei, Z., Wang, S.: Learning spatiotemporal representations for human fall detection in surveillance video pp. 215–230.<https://doi.org/10.1016/j.jvcir.2019.01.024>
14. Kwolek, B., Kepski, M.: Human fall detection on embedded platform using depth maps and wireless accelerometer (3), 489–501.<https://doi.org/10.1016/j.cmpb.2014.09.005>
15. M, P.V., Shekar, M., Pragathi B, S.L., Ngadiran, R., Ravindran, S.: Fall detection system for monitoring elderly people using yolov7-pose detection model. In: 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3). pp. 1–6 (2023).<https://doi.org/10.1109/IC2E357697.2023.10262506>

16. 5. Montero-Odasso, M., Van Der Velde, N., Martin: World guidelines for falls prevention and management for older adults: A global initiative (9), afac205.<https://doi.org/10.1093/ageing/afac205>
17. Nogas, J., Khan, S.S., Mihailidis, A.: DeepFall: Non-Invasive Fall Detection with Deep Spatio-Temporal Convolutional Autoencoders. *Journal of Healthcare Informatics Research* 4(1), 50–70 (2019). <https://doi.org/10.1007/s41666-019-00061-4>
18. Noor, N., Park, I.K.: A lightweight skeleton-based 3D-CNN for real-time fall detection and action recognition. In: 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 2171–2180. IEEE.<https://doi.org/10.1109/ICCVW60793.2023.00232>
19. Saha, B., Islam, M.S., Kamrul Riad, A., Tahora, S., Shahriar, H., Sneha, S.: Blockthefall: Wearable device-based fall detection framework powered by machine learning and blockchain for elderly care. In: 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). pp. 1412–1417.<https://doi.org/10.1109/COMPSAC57700.2023.00216>
20. Saleh, M., Abbas, M., Le Jeannès, R.B.: FallAllID: An open dataset of human falls and activities of daily living for classical and deep learning applications (2), 1849–1858.<https://doi.org/10.1109/JSEN.2020.3018335>
21. Saleh, M., Abbas, M., Prud’Homm, J., Somme, D., Le Bouquin Jeannes, R.: A reliable fall detection system based on analyzing the physical activities of older adults living in long-term care facilities pp. 2587–2594.<https://doi.org/10.1109/TNSRE.2021.3133616>
22. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks (Oct 2015), <http://arxiv.org/abs/1412.0767>
23. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6450–6459. IEEE, Salt Lake City, UT (Jun 2018).<https://doi.org/10.1109/CVPR.2018.00675>
24. Vishnu, C., Datla, R., Roy, D., Babu, S., Mohan, C.K.: Human fall detection in surveillance videos using fall motion vector modeling (15), 17162–17170.<https://doi.org/10.1109/JSEN.2021.3082180>
25. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7464–7475 (2023).<https://doi.org/10.1109/CVPR52729.2023.00721>
26. Wang, F., Liu, J., Hu, G.D.: A novel indoor human fall detection method based on an end-to-end neural network and bagged tree classifier. In: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence. p. 384–389. ACAI ’19, New York, NY, USA (2020).<https://doi.org/10.1145/3377713.3377767>, <https://doi.org/10.1145/3377713.3377767>
27. Wang, L., Koniusz, P., Huynh, D.: Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8697–8707 (2019).<https://doi.org/10.1109/ICCV.2019.00879>
28. Wang, S., Li, Z., Zhao, Y., Xiong, Y., Wang, L., Lin, D.: denseflow. <https://github.com/open-mmlab/denseflow> (2020)
29. Wang, X., Jia, K.: Human fall detection algorithm based on yolov3. In: 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC). pp. 50–54.<https://doi.org/10.1109/ICIVC50857.2020.9177447>

30. World Health Organization: Fact sheet: Falls (2021), <https://www.who.int/zh/news-room/fact-sheets/detail/falls>, last accessed 31 May 2024
31. Wu, L., Huang, C., Fei, L., Zhao, S., Zhao, J., Cui, Z., Xu, Y.: Video-based fall detection using human pose and constrained generative adversarial network pp. 1–1. <https://doi.org/10.1109/TCSVT.2023.3303258>
32. Xu, D., Lu, X.: Fall- lstm: A fall detection network based on spatio-temporal location module. In: 2023 42nd Chinese Control Conference (CCC). pp. 8707–8714. <https://doi.org/10.23919/CCC58697.2023.10240325>
33. Yadav, S.K., Luthra, A., Tiwari, K., Pandey, H.M., Akbar, S.A.: Arfdnet: An efficient activity recognition & fall detection system using latent feature pooling p. 107948. <https://doi.org/10.1016/j.knosys.2021.107948>
34. Zhou, C., Xiao, J., Xiong, A., Zhang, C.: Human fall detection based on improved particle swarm optimization algorithm and neural network. In: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA). pp. 1–4. <https://doi.org/10.1109/CVIDLICCEA56201.2022.9823997>



# Synthetic Video Generation for Weakly Supervised Cross-Domain Video Anomaly Detection

Pradeep Narwade<sup>1</sup>, Ryosuke Kawamura<sup>2(✉)</sup>, Gaurav Gajbhiye<sup>1</sup>,  
and Koichiro Niinuma<sup>2</sup>

<sup>1</sup> Fujitsu Consulting India Private Limited, Pune, India

<sup>2</sup> Fujitsu Research of America, Pittsburgh, PA, USA

rkawamura@fujitsu.com

**Abstract.** Video anomaly detection (VAD) plays a pivotal role in crucial applications such as security and surveillance, garnering significant interest from the research community. The utility of cross-domain VAD is critical in practical scenarios, yet most of research remains focused on same-domain VAD. Weakly supervised approaches excel in same-domain contexts but are rarely applied to cross-domain VAD, which typically relies on unsupervised methods. This paper presents a new weakly supervised framework for addressing cross-domain VAD challenges, aiming to improve model generalization across different domains. A key issue is the model's propensity for overfitting to source domain anomalies, impairing its ability to detect out-of-distribution anomalies. Our approach introduces a video synthesis technique using generative technologies for zero-shot cross-domain VAD. This strategy combats the generative technologies' limitations, especially their struggle to generate human behavior and object motion accurately—vital aspects of VAD. By merging generative video editing with object synthesizing, we ensure that synthesized videos maintain their original normal or abnormal status. Combining synthesized with original data, our model is trained in a weakly supervised manner. The experimental results demonstrate that our method outperforms existing works for cross-domain scenarios.

**Keywords:** Video Anomaly Detection · Zero-Shot Cross-Domain Video Anomaly Detection · Data Augmentation · Synthesis of Anomaly Videos · Weakly Supervised Learning

## 1 Introduction

Video anomaly detection (VAD) is a computer vision task that involves identifying events that are unexpected to occur, such as fighting and shoplifting. VAD has practical applications in a wide range of fields, including security and surveillance fields. A critical aspect of VAD is cross-domain capability, where

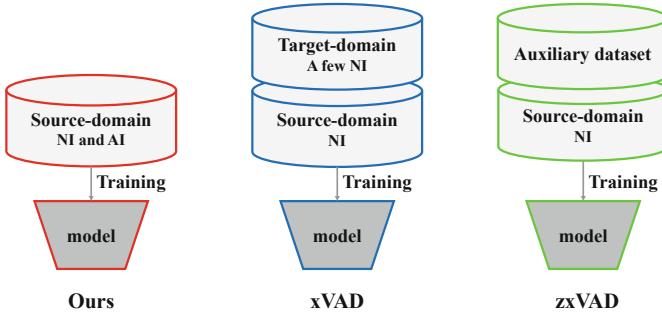
---

P. Narwade and R. Kawamura—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15315, pp. 375–391, 2025.

[https://doi.org/10.1007/978-3-031-78354-8\\_24](https://doi.org/10.1007/978-3-031-78354-8_24)



**Fig. 1.** Comparison of cross-domain VAD problem settings: our methodology vs. existing approaches. ‘NI’ stands for normal instances and ‘AI’ stands for anomalous instances. Traditional cross-domain VAD models (xVAD [15, 17] and zxVAD [2]) typically depend on source domain normal data supplemented with information from external datasets, such as select instances from the target domain or auxiliary datasets. In contrast, our approach exclusively uses normal and anomalous data from the source domain for model training.

a model trained on a source domain is tested on a target domain. This is crucial due to variations in context, location, camera angles, and types of anomalies between domains, making it impractical and costly to tailor models for each specific context. Despite its importance, most of the existing research concentrates on same-domain VAD, leaving cross-domain VAD relatively unexplored. Current cross-domain VAD studies primarily utilize unsupervised methods, focusing on normal behavior and identifying anomalies as deviations [2, 15, 17]. However, while weakly supervised methods show superior performance in same-domain scenarios, their potential in cross-domain contexts remains untapped. This paper introduces a novel approach for weakly supervised cross-domain VAD, leveraging insights from weakly supervised techniques in same-domain settings to enhance cross-domain performance. Figure 1 illustrates the distinctions between our problem setting and those of existing approaches. For instance, as a general expectation, our cross-domain WSVAD approach aims to enable models trained on limited intersections to be effectively deployed across diverse settings, such as other intersections, footpaths.

One challenge in weakly supervised cross-domain VAD is the tendency to overfit source domain anomalies, diminishing the ability to recognize out-of-distribution anomalies. This overfitting arises because weakly supervised methods are trained on both normal and anomalous events within the source domain, potentially leading them to recognize only the anomaly types present in that dataset.

To address this, we introduce a synthetic video generation technique for cross-domain VAD within a weakly supervised framework. This method is designed to enrich source domain data diversity and aims to prevent overfitting by employing advanced generative technologies for creating or editing videos based on given

prompts. This data augmentation solely utilizes the videos provided by the original training set from the source domain. Our data augmentation approach aims to alter only the style of the videos from the training set in the source domain, without changing the movements of humans or objects. This ensures that videos synthesized from normal ones remain normal while those synthesized from abnormal ones remain abnormal.

However, while state-of-the-art video-to-video transformation approaches have great ability, they often fall short in accurately depicting human behaviors or object movements, which are crucial elements in VAD. Our solution combines generative and object segmentation techniques to produce videos with realistic human and object movements. We utilize Segment-and-Track-Anything (SAM-Track) technology [6] to extract objects from videos, integrating them with content generated by TokenFlow [9] to construct new training materials based on the UR-DMU modules [29], which represent one of the state-of-the-art approaches for same-domain VAD. Unlike traditional cross-domain VAD methods that rely on unsupervised learning, our approach does not necessitate data from the target domain or any supplementary datasets. To adapt UR-DMU for cross-domain VAD, we have modified the original model in two key ways: 1) Our method uses ViCLIP [25] for feature extraction instead of I3D [26]; 2) Our method employs a dense Multi-Layer Perceptron (MLP) for the classification header. The effectiveness of our approach is validated through experiments in cross-domain settings, including ShanghaiTech [16] and Avenue [14] datasets. Our contributions are as follows:

- Introduction of a cross-domain anomaly detection method in a weakly supervised framework, which integrates a modified UR-DMU model with new feature extraction via ViCLIP and a dense MLP classification header.
- Development of a unique synthetic video generation technique to enhance the resilience of the weakly supervised framework against overfitting. This technique comprises two phases: style editing using TokenFlow and object synthesizing with SAM-Track.
- Demonstration of our method’s efficacy in cross-domain VAD scenarios.

## 2 Related Works

### 2.1 Weakly Supervised VAD

The task of weakly supervised video anomaly detection (WSVAD) is crucial in the Computer Vision domain, recognized for its practicality and precision. This approach has prompted the exploration of various methodologies to address the challenge. In contrast to unsupervised VAD, which does not necessitate annotated anomaly data but encounters difficulties in anomaly detection, WSVAD employs video-level annotations. This system labels entire videos as normal or abnormal, significantly reducing the effort required for data collection.

The Multiple Instance Learning framework is a cornerstone and effective strategy for WSVAD [20]. Numerous approaches have been proposed: RTFM [24]

uses dilated convolutions and self-attention mechanisms to discern feature magnitudes and grasp temporal dependencies across ranges. Joo et al. [11] utilize the CLIP [22] image encoder for extracting features and employ temporal self-attention for temporal dependency analysis. Li et al. [13] propose a Multi-Sequence Learning (MSL) approach with a Transformer-based architecture for assessing anomaly scores at both video and snippet levels, further enhancing accuracy through a self-training refinement strategy. Majhi et al. [19] innovate with the Outlier-Embedder and Cross Temporal Scale Transformer, aimed at understanding the temporal interplay between anomalies and normal events, and capturing global temporal relations, respectively. Zhou et al. [29] introduce an uncertainty learning model complemented by a global-local multi-head self-attention module for effective spatial and temporal feature integration, enriched with a memory unit for feature distinction. While these techniques have demonstrated their efficacy in same-domain WSVAD applications, their performance in cross-domain contexts is yet to be explored.

## 2.2 Cross-Domain VAD

The task of VAD within same-domain settings has been extensively explored in many studies (see [20] for details). In contrast, only a handful studies, such as those mentioned in [2, 15, 17], have focused on cross-domain settings for VAD. In cross-domain evaluation, the test dataset is collected under an entirely different environment from the training dataset. This approach enables the assessment of VAD performance across different contexts and domains, which is critically important for practical applications in terms of its adaptability and robustness.

Studies [15, 17] utilize few-shot learning approaches to adopt the context of the target domain. Although effective, these methods require a small quantity of target domain data. Unlike these methods, the technique introduced by Aich et al. [2] requires no data from the target domain. Instead, their method generates pseudo-anomaly instances by superimposing humans from an auxiliary dataset, such as those used for action recognition tasks. The findings in [2] demonstrate superior performance over methods based on the few-shot approach in cross-domain scenarios, even though it still relies on external datasets. Previous strategies for cross-domain anomaly detection depend on either data from the target domain or supplementary datasets to improve detection capabilities in novel contexts.

These approaches employed unsupervised methods, while weakly supervised approaches are generally used in same-domain scenarios. This is advantageous in terms of avoiding overfitting to specific type of anomalous events, but it also means that they cannot leverage insights from the well-studied weakly supervised approaches. Our study diverges from these methods by utilizing weakly supervised frameworks. The key to our approach is enhancing data solely using source domain resources, without the need for any additional datasets, to avoid overfitting.

### 2.3 Synthetic Image for VAD

Data augmentation based on synthetic data for VAD has been introduced in previous studies. [4, 21] utilize generative model such as GAN and Auto Encoder to produce pseudo-anomaly data. [10] proposes clustering-based pseudo-anomaly generation, where normal data is separated into k-clusters and one of the clusters is treated as normal and the others as abnormal. The method proposed in [5] generates irregular motions for each object separately and uses the pseudo motion for self-training scheme. A cut-and-paste strategy is used in [1-3] to produce pseudo-anomaly data. These methods are unsuitable for direct application within WSVAD frameworks, as they generally generate pseudo anomalies that appear in all frames of the generated videos. Unlike these previous methods for synthetic image generation for VAD tasks, our data augmentation approach can be directly applied to WSVAD frameworks.

## 3 Methods

Initially, we introduce our data augmentation framework in Fig. 2, which generates synthetic normal and abnormal videos using only the source domain’s videos. This approach utilizes advanced generative techniques to minimize inaccuracies in human or object movements common in generated videos.

The framework consists of two components: Style Editing, which employs a video editing technique to produce a variety of video style, and Object Synthesizing, aimed at maintaining the visual integrity of the foreground objects. Our framework is capable of generating synthetic video data independently, without the need for any additional datasets. It specifically targets the generation of synthetic videos depicting person-based anomalous activities.

Furthermore, we introduce a novel weakly supervised learning architecture for video anomaly detection. This architecture utilizes spatial and temporal feature extractors to derive robust snippet features, which are then refined by GLMHSA and DMU modules [29]. The processed features are subsequently fed into our advanced classification modules.

In the following sections, we delve into the details of the video augmentation framework, and describe our video anomaly detection architecture.

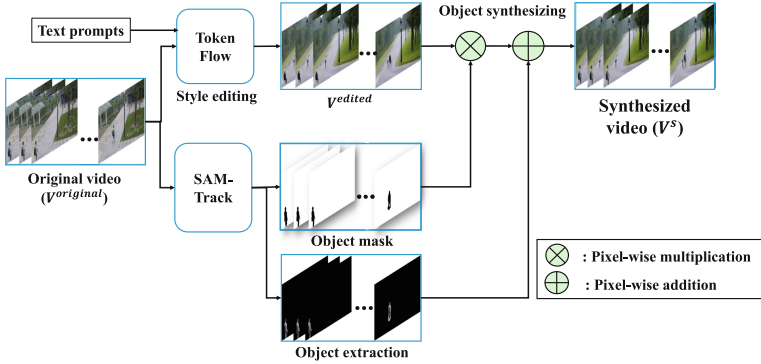
### 3.1 Synthetic Video Generation

Our generation scheme consists of two main components: Style Editing and Objects Synthesizing.

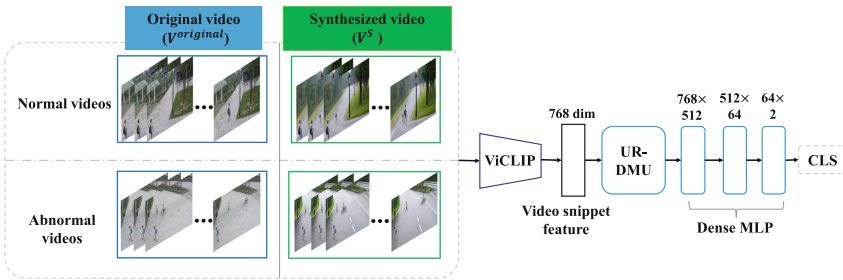
**Style Editing.** To enhance the diversity of videos, we emphasize the manipulation of style information. The style of video can significantly impact event recognition within videos, and our goal is to mitigate these effects by increasing style diversity. For this purpose, we utilize TokenFlow [9], a text-guided video editing framework. TokenFlow facilitates seamless video modifications through



### 1. Data Augmentation



### 2. Learning with Augmented data



**Fig. 2.** Pipeline of the proposed approach. This methodology consists of two primary phases: 1) Data Augmentation, which leverages synthetic video generation, and 2) Learning, where the model is trained using the augmented dataset. The augmentation phase incorporates Style Editing and Object Synthesizing techniques, seamlessly integrating style-edited frames with extracted humans and objects. In the learning phase, the model undergoes training with both the original data from the source domain and the synthetically augmented data. The ViCLIP [25] model extracts 768-dimensional features from each video snippet, which are then forwarded to the UR-DMU block [29]. The outputs of this block serve as inputs to a three-layered Dense MLP.

a pre-trained text-to-image diffusion model, eliminating the need for additional training or fine-tuning. A source dataset video  $V^{original}$  is processed by Token-Flow, using prompts to specify desired style, resulting in an style-edited video denoted as  $V^{edited}$ .

**Object Synthesizing.** To preserve the integrity of foreground elements in the original videos, objects and individuals are extracted and then superimposed onto the style-edited video  $V^{edited}$ . This step is crucial as the video editing process may inadvertently alter or obscure the appearance of these foreground elements. For this purpose, we employ the SAM-Track model [6] to identify object regions in the source videos. First, we extract a sequence of frame-level object masks,  $I_i^M$  from the  $i$ th frame of  $V^{original}$  using the SAM-Track model.

Second, The actual objects in the original video frame is then isolated by applying the mask through element-wise multiplication of the  $i$ th frame’s mask ( $I_i^M$ ) and  $i$ th original frame ( $I_i^{V^{original}}$ ), denoted as  $I_i^O = I_i^M \otimes I_i^{V^{original}}$ , where  $\otimes$  represents multiplication. We also create a sequence of inverted mask frames ( $\overline{I_i^M}$ ) for frame-level blending. Finally, a synthesized video ( $V^S$ ) is created.  $I_i^{V^S}$ , the  $i$ th frame of  $V^S$ , is produced by combining the extracted objects with the frame generated by TokenFlow ( $I_i^{V^{edited}}$ ) using the inverted mask. This process is formulated as  $I_i^{V^S} = I_i^O \oplus (\overline{I_i^M} \otimes I_i^{V^{edited}})$ , where  $\oplus$  denotes pixel-wise addition. These synthesized and original videos  $V^{original}$  and  $V^S$ , respectively, are utilized to learn robust feature representations. Figure 3 presents a comparison between video frames with and without the object synthesizing. As illustrated in Fig. 3, our technique successfully integrates foreground objects from the original source video into the style-edited video, despite occasional distortions or disappearances of person body parts in the edited videos.

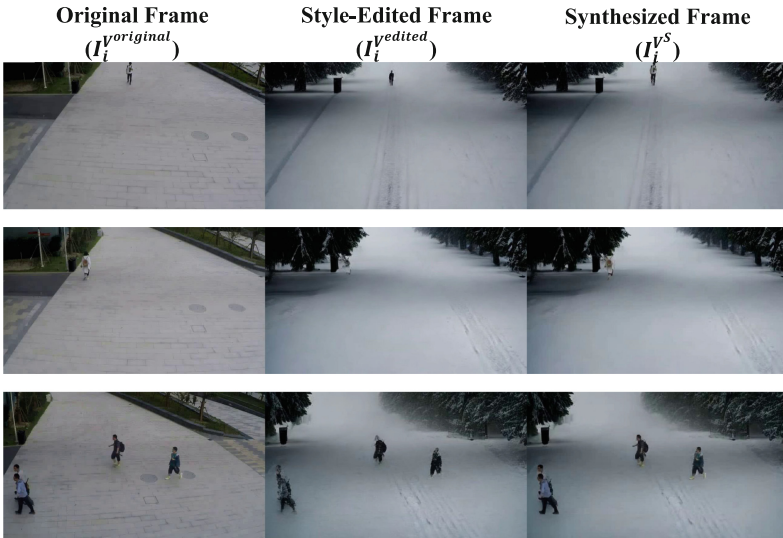


Fig. 3. The comparison of with and without object synthesizing

### 3.2 Constructing Video Anomaly Detection Modules

To begin, we define the problem settings for WSVAD. Each video from sets  $V^{original}$  and  $V^S$  is divided into multiple snippets and we denote each video snippet  $v \in \mathbb{R}^{F \times C \times H \times W}$  ( $F$  is the number of frames,  $C$  is the number of channels,  $H$  is the height and  $W$  is the width of frames). We follow a Multiple-Instance Learning framework [20]. Within this framework, a video is labeled as  $y = 1$  if it contains at least one abnormal snippet, and  $y = 0$  if all snippets are normal,

although normal snippets may also be present in videos labeled as  $y = 1$ . The objective of WSVAD is to develop a model capable of generating an anomaly score  $S = f(v)$  for each snippet.

We construct a video anomaly detection model using both original source videos and synthesized videos within a weakly supervised framework. Our architecture comprises two main modules: a Video Snippet Feature Extractor and an Anomaly Classifier. Initially, in line with the typical process for WSVAD, video features from each snippet are extracted using existing pretrained models. For our module, we use ViCLIP [25], a variant of CLIP [22] fine-tuned for video processing, as the video snippet feature extractor. ViCLIP [25] adapts video-text representations, significantly enhancing video understanding tasks.

We further employ the UR-DMU modules which is the integration of Global and Local Multi-Head Self Attention (GL-MHSA), Dual Memory Units (DMU), and Normal Data Uncertainty Learning (NUL) modules [29] together with a new denser Multi-Layer Perceptron than the original one, to serve as our anomaly classifier. These modules, GL-MHSA, DMU, and NUL, are specialized feature embedding systems crafted to distinguish between normal and abnormal features within video snippets. GL-MHSA is adept at identifying both immediate and extended temporal correlations within anomalous features. Meanwhile, DMU archives feature prototypes from standard data, and NUL is tasked with mapping the distribution of such normal data. The synthesized features from GL-MHSA and DMU are channeled into the classification layer. For a nuanced analysis of features from diverse datasets, we deploy a deeply layered and denser MLP, which consists of three layers of MLP. Regarding loss functions, we adopt the loss functions as outlined by Zhou et al. [29], which include Binary Cross Entropy Loss for classification, dual memory loss for differentiating normal and abnormal patterns, triplet loss to separate these features, Kullback-Leibler diversity regularization for stable learning, and distance loss to enhance the margin between normal and abnormal features

## 4 Experiments

### 4.1 Datasets

Following existing works for cross-dataset VAD, we evaluated our approach under a cross-domain scenario, trained on the SHT dataset [16] and tested on the Avenue dataset [14] and Ped2, as detailed in Sect. 4.3 and Sect. 4.7, respectively. With advancements in image sensor technology, high-resolution RGB color cameras have become prevalent in surveillance, making the evaluation of VAD tasks on RGB-colored, high-resolution videos increasingly crucial. We did not use the UCF-Crime [23], Ped1 [18] datasets for our evaluation due to their low resolution and low clarity, which are  $240 \times 320$ ,  $158 \times 238$ , respectively. The UCF-Crime dataset, sourced from the internet, is particularly noted for its low-quality videos. Such limitations in resolution and clarity are typical reasons for their exclusion in related research studies [8, 10].

**ShanghaiTech (SHT)** [16]: The ShanghaiTech dataset comprises 437 videos, with 307 normal and 130 anomaly videos such as riding a bicycle, crossing a road, and jumping forward across 13 different settings. Each video is presented at a resolution of  $480 \times 856$ . It is predominantly utilized for unsupervised anomaly detection, as the training set exclusively contains normal events. The footage is captured from elevated angles. Additionally, a revised version for weakly supervised learning, termed SHT-V2, was introduced by [28], incorporating a subset of anomaly videos from the original test set into the training set.

**Avenue** [14]: Collected at the CUHK campus Avenue, this dataset includes 16 training and 21 test videos, featuring a total of 47 abnormal events such as running, irregular walking directions, cycling, and throwing objects. Each video is presented at a resolution of  $360 \times 640$ .

**UCSD Ped2** [18]: A compact video anomaly dataset, UCSD Ped2 consists of 28 videos, including 16 for training and 12 for testing. It catalogs anomalies like running, biking, and skateboarding. Unlike SHT-V2 and Avenue, the Ped2 dataset are grayscale videos and low resolution ( $240 \times 360$ ).

## 4.2 Evaluation Settings

**Implementation Details.** Each video is segmented into snippets of 16 frames each, with a resolution of  $224 \times 224$  pixels. For snippet feature extraction, we employ the ViCLIP [25] model, which has been pretrained on dataset InternVid-10M-FLT [25]. We have chosen eight evenly spaced frames out of sixteen and passed them through ViCLIP [25] to extract features. We utilize the URDMU module, excluding its original classification head, while retaining the same parameters as mentioned in [29]. Instead of the original classification head, we introduce a three-layer MLP as the new classification head. The original classification head consists of two layers with output nodes 128 and 1. In our classification head, the first, second, and third layers are designed with output nodes 512, 64, and 1, respectively. The learning rate is configured at 0.0001, utilizing the Adam optimizer, with a batch size of 32.

In our approach, we enhance the training dataset by creating two distinct style videos for each original video, effectively tripling the size of the training dataset. We employed “road, person, building, trees, snowfall” and “road, person, building, trees, rainy-day” as prompts for style editing in data augmentation. TokenFlow aims to modify videos to match specific input text prompts. However, selecting suitable prompts is essential to achieve realistic outcomes; otherwise, the results may be impractical. We experimented with various prompts on TokenFlow using videos from our training dataset, with outcomes displayed in Fig. 4. Notably, in the video edited with the prompt “persons walking on street” (see Fig. 4, right), stationary individuals were generated, making their presence appear unnatural despite the normal behavior of people in the original video. These findings suggest that although TokenFlow holds promise, there’s a notable risk of generating unsatisfactory or unrealistic videos without meticulous prompt selection. In this paper, we carefully selected prompts for style editing to minimize artifacts.

For the cross-domain VAD, our model is trained using the training set from the source domain (SHT-V2) and assessed on the test sets of target domains (Avenue). The macro and micro Area Under the Curve (AUC) are employed to gauge performance. It should be noted that existing cross-domain VAD methods did not calculate micro AUC scores; therefore, we only present their macro AUC scores for comparison.



Fig. 4. Examples of outputs generated by TokenFlow using various prompts

### 4.3 VAD Results of Our Proposed Approach on Cross-Domain Datasets

Table 1 reports the AUC scores of our proposed method on the test set of cross-domain settings as well as scores of our method without data augmentation and original UR-DMU baseline. We should note that existing methods for cross-domain scenarios also employ SHT-V1 dataset, which is designed for unsupervised learning manner and is different from SHT-V2 dataset.

In the Avenue Test dataset, our method achieves a higher macro AUC score (83.42%) than existing methods for cross-domain VAD and original UR-DMU baseline. In addition, our method with augmented data marks a higher macro AUC score than the original UR-DMU (78.90%) and our method without augmented data (68.60%). In terms of micro AUC, our method (88.92%) also outperformed the original UR-DMU (86.38%) and ours without augmented data (77.36%). These results suggest that our method effectively improves the performance of VAD in cross-domain dataset settings. Ours without augmented data scores lower than the original UR-DMU. This outcome is likely due to our architecture incorporating a denser classification head, which requires more data for effective training.

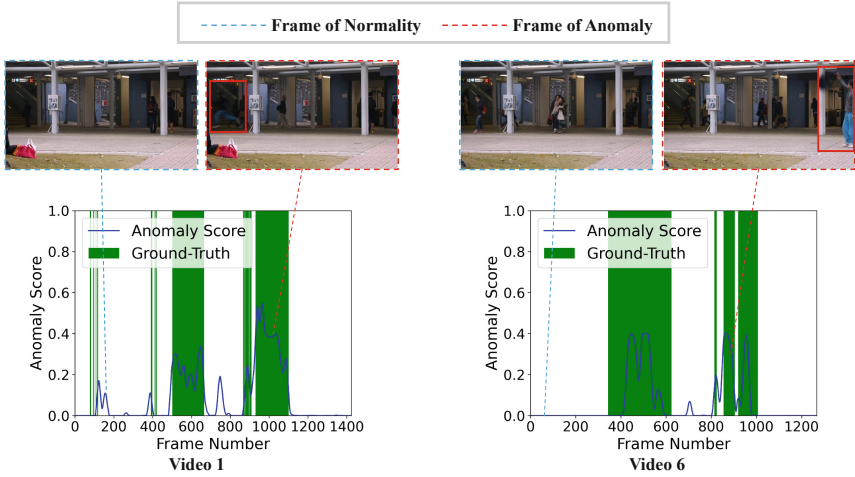
Figure 5 shows the frame-level anomaly scores for videos in the Avenue dataset. The results show that our method detects anomaly events precisely.

### 4.4 Ablation Studies

**Impact of Object Synthesizing on Anomaly Detection.** We explore the effect of adding an object synthesizing phase (Sect. 3.1) on Video Anomaly Detection effectiveness. In our assessment, we leveraged the TokenFlow output,  $V_{bg}$ , as

**Table 1.** Comparison of AUC scores with other methods. The best score is marked in bold font and the second is marked in underline. ‘U’ stands for Unsupervised Learning and ‘WS’ stands for Weakly Supervised Learning.

Learning	Method	macro AUC	micro AUC
U	rGAN [15]	71.43	—
	MPN [17]	74.06	—
	zxVAD [2]	<u>83.19</u>	—
WS	UR-DMU (our implementation) [29]	78.90	<u>86.38</u>
	Ours (w/o Augmented data)	68.60	77.36
	Ours (with Augmented data)	<b>83.42</b>	<b>88.92</b>



**Fig. 5.** Anomaly score on test sample Video 01 (left) and Video 06 (right) from Avenue

augmented data, foregoing the application of object synthesis. The performance comparison of VAD with and without the object synthesizing phase on Avenue dataset is detailed in Table 2. The data in Table 2 show that incorporating the object synthesizing phase yields a performance score of 83.42% (macro AUC) and 88.92% (micro AUC), markedly surpassing the macro AUC of 61.12% and micro AUC of 71.02% obtained without the object synthesizing step. This finding highlights the crucial enhancement the object synthesizing phase brings to the efficiency of Weakly supervised VAD across different domains.

**Effect of Feature Extractors.** The choice of video snippet feature extractor significantly impacts the accuracy of video anomaly detection, as highlighted by existing studies. To evaluate the effect of different feature extractors, we conducted a comparative experiment using extractors highlighted in prior research:

**Table 2.** Comparison of AUC scores with and without Object Synthesis

Method	macro	micro
Ours w/o Object Synthesizing	61.12	71.02
Ours	83.42	88.92

I3D [26], S3D [27], and CLIP [22]. These extractors are widely recognized in various Video Anomaly Detection (VAD) studies, with I3D and S3D commonly employed as video snippet feature extractors. The application of CLIP [22] for the VAD task was introduced by Joo et al. [11], from whom we adopted the pre-trained model for our analysis. The experiment’s results, represented as AUC scores, are detailed in Table 3. According to Table 3, the macro AUC scores for I3D, S3D, and CLIP are 73.24%, 75.82%, and 62.21%, respectively, while micro AUC scores are 84.37%, 85.26%, and 77.24% each. In contrast, ViCLIP [25] demonstrates superior performance, achieving macro AUC of 83.42% and micro AUC of 88.92% on Avenue, thus surpassing the other feature extractors. These findings underscore ViCLIP’s superior capability in extracting more effective features for the Weakly Supervised VAD task. It is beneficial to note that applying ViCLIP [25] to the methods [2, 15, 17] listed in Table 1 is not feasible, as these existing methods utilize auto-encoders for individual frame analysis in an unsupervised learning context and are not designed to handle temporal information.

**Table 3.** Comparison of AUC scores across different video snippet feature extractors

Feature Extractor	macro	micro
I3D	73.24	84.37
S3D	75.82	85.26
CLIP	62.21	77.24
ViCLIP [25](Ours)	83.42	88.92

**Effect of the Dense MLP.** In our approach, we incorporated dense MLP as a new classification head for the UR-DMU modules. To validate the efficacy of employing dense MLP for classification, we carried out an experiment. In this experiment, augmented data was utilized to train both the version with the dense MLP and the version with the original classification head. Table 4 presents a comparison of AUC scores between our dense MLP and the original classification head as used in [29]. The AUC scores achieved with the dense MLP (83.42% of macro AUC and 88.92% of micro AUC) surpassed those obtained with the original classification head in [29] (68.69% of macro AUC and 83.10% of micro AUC). These findings demonstrate the enhanced performance of our dense MLP in the context of cross-domain weakly supervised VAD tasks.

**Table 4.** Comparison of AUC scores with our Dense MLP and original classification head

Classification head	macro	micro
Original classification head [29]	68.69	83.10
Dense MLP	83.42	88.92

#### 4.5 Effect of Augmented Data on Other WSVAD Methods

To assess the impact of our augmented data on other Weakly Supervised Video Anomaly Detection (WSVAD) methods, we conducted a series of experiments. In these experiments, we integrated our augmented data into various WSVAD methods and compared their performance with and without the augmented data. The results of these comparisons are detailed in Table 5. The performance of RTFM and BN-WVAD declines with our augmented data, likely due to their designs being optimized for smaller datasets, which may not effectively handle extensive augmentation. Conversely, our method benefits significantly from augmented data. Based on the results in Table 4, we can assume that this improvement is largely due to the efficacy of Dense MLP in processing large volumes of data.

**Table 5.** Comparison of performance among various WSVAD methods with and without our augmented data

Method	macro AUC
RTFM (w/o Augmented data) [24]	70.37
RTFM (w/ Augmented data) [24]	59.26
BN-WVAD (w/o Augmented data) [30]	61.47
BN-WVAD (w/ Augmented data) [30]	52.97
UR-DMU (w/o Augmented data) [29]	78.90
UR-DMU (w/ Augmented data) [29]	80.07
Ours (w/o Augmented data)	68.60
Ours (with Augmented data)	<b>83.42</b>

#### 4.6 Same-Domain Evaluation

Table 6 displays the AUC scores from the same-domain evaluation on the SHT-V2 dataset. The table reveals that our augmented model yields an increase in AUC scores by 1.94% and 0.14% over the baseline UR-DMU model and our approach without augmentation, respectively. This evidence underscores the efficacy of our method for the Video Anomaly Detection (VAD) task within the same-domain context. Compared to existing state-of-the-art methods [7, 12, 24] that



focus only on same-domain scenarios, our method achieves comparable scores despite targeting cross-domain scenarios.

**Table 6.** Comparison of AUC Scores between Ours and existing methods in a same-domain scenario, trained and tested on the SHT-V2 Dataset. The best score is marked in bold font and the second is marked in underline.

Method	macro	micro
RTFM [24]	97.21	–
SSRL [12]	<b>97.98</b>	–
Cho et al. [7]	97.60	–
BN-WVAD [30]	<u>97.61</u>	–
UR-DMU (our implementation)	94.55	<u>99.06</u>
Ours (w/o Augmented data)	96.35	98.16
Ours (with Augmented data)	96.49	<b>99.24</b>

#### 4.7 Evaluation on Low-Resolution Grayscale Dataset

To assess the potential effectiveness of our approach in low-quality video, we conducted experiments with the UCSD Ped2 dataset. It’s important to note that, given the current state of surveillance technology, the relevance of evaluating VAD on low-resolution, grayscale, and small datasets is diminishing. In this experiment, the models are trained on SHT-V2 dataset and tested on Ped2 dataset. Considering gray-scaled data, we evaluate both RGB and gray-scale data augmentation.

**Table 7.** Comparison of AUC scores with other methods on UCSD Ped2 datasets. The best score is marked in bold font and the second is marked in underline. ‘U’ stands for Unsupervised Learning and ‘WS’ stands for Weakly Supervised Learning.

Learning	Method	macro	micro
U	rGAN [15]	81.95	–
	MPN [17]	90.17	–
	zxVAD [2]	<b>95.80</b>	–
WS	UR-DMU (our implementation) [29]	83.89	–
	Ours (w/o Augmented data)	90.22	91.72
	Ours (with Augmented data (RGB))	90.46	<u>91.91</u>
	Ours (with Augmented data (Gray-scale))	<u>93.43</u>	<b>92.65</b>

The results are shown in Table 7. On the Ped2 Test dataset, our method recorded macro AUC scores of 93.43% for grayscale and 90.46% for RGB, with

micro AUC scores of 92.65% for grayscale and 91.91% for RGB, which are higher than ours without augmented data (macro AUC of 90.22% and micro AUC of 91.72%). The UCSD Ped2 dataset, characterized by its lower resolution and quality, is a significant factor contributing to the underperformance observed in our evaluations. While our results on this dataset are modestly lower compared to zxVAD [2], which reported 95.80% accuracy, we primarily attribute this discrepancy to the difference in video quality between the source dataset, including the augmented data, and the target domain dataset. In our data augmentation process, we generate videos with higher resolution and clarity. Conversely, the zxVAD approach as described in [2] utilizes images of varying quality, including some with notably low clarity. In practice, our method with grayscale data augmentation outperformed our method with RGB data augmentation, indicating that domain-specific data augmentation can markedly enhance cross-domain VAD performance.

#### 4.8 Computational Cost

We assessed the computational cost of UR-DMU, our method with I3D (Ours(I3D)), and our method using ViCLIP (Ours(ViCLIP)) by calculating frames per second (FPS) and the number of model parameters, following the methodology outlined in [2]. This evaluation was conducted using a single A100 GPU. Both UR-DMU and Ours(I3D) share similar components and architectures, with the primary distinction being the integration of Dense MLP in Ours(I3D). In terms of performance, Ours(ViCLIP) operates at 6.69 FPS, while Ours(I3D) achieves 39.20 FPS, and UR-DMU records 39.26 FPS. Despite Ours(ViCLIP) being slower, it maintains a speed that is feasible for practical applications. Regarding model complexity, Ours(ViCLIP) has 310,500,482 parameters, significantly higher than Ours(I3D) with 19,616,561 parameters, and UR-DMU with 19,190,193 parameters. These figures illustrate the trade-offs between computational efficiency and model complexity in our designs.

## 5 Conclusion

This paper tackles the challenge of weakly supervised Video Anomaly Detection in cross-domain scenarios. We introduce a novel data augmentation strategy leveraging synthetic image generation tailored for weakly supervised learning contexts. Our weakly-supervised learning framework features a ViCLIP-based feature extractor, the UR-DMU module, and an innovative dense MLP classifier. This ensemble is designed to effectively discern normal and anomalous patterns based solely on weak labels. The experimental findings support the effectiveness of our approach. In future work, we plan to evaluate our method across multiple datasets and explore various prompts for style editing.

## References

1. Acsintoae, A., et al.: Ubnormal: new benchmark for supervised open-set video anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20143–20153 (2022)
2. Aich, A., Peng, K.C., Roy-Chowdhury, A.K.: Cross-domain video anomaly detection without target domain adaptation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2579–2591 (2023)
3. Astrid, M., Zaheer, M.Z., Lee, J.Y., Lee, S.I.: Learning not to reconstruct anomalies. In: BMVC (2021)
4. Astrid, M., Zaheer, M.Z., Lee, S.I.: Synthetic temporal anomaly guided end-to-end video anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 207–214 (2021)
5. Barbalau, A., et al.: Ssmtl++: revisiting self-supervised multi-task learning for video anomaly detection. *Comput. Vis. Image Understand.* 103656 (2023)
6. Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., Yang, Y.: Segment and track anything. arXiv preprint [arXiv:2305.06558](https://arxiv.org/abs/2305.06558) (2023)
7. Cho, M., Kim, M., Hwang, S., Park, C., Lee, K., Lee, S.: Look around for anomalies: weakly-supervised anomaly detection via context-motion relational learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12137–12146 (2023)
8. Doshi, K., Yilmaz, Y.: A modular and unified framework for detecting and localizing video anomalies. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3982–3991 (2022)
9. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: Tokenflow: consistent diffusion features for consistent video editing. arXiv preprint [arXiv:2307.10373](https://arxiv.org/abs/2307.10373) (2023)
10. Ionescu, R.T., Khan, F.S., Georgescu, M.I., Shao, L.: Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7842–7851 (2019)
11. Joo, H.K., Vo, K., Yamazaki, K., Le, N.: Clip-tsa: clip-assisted temporal self-attention for weakly-supervised video anomaly detection. In: 2023 IEEE International Conference on Image Processing, pp. 3230–3234 (2023)
12. Li, G., Cai, G., Zeng, X., Zhao, R.: Scale-aware spatio-temporal relation learning for video anomaly detection. In: Proceedings of the European Conference on Computer Vision, pp. 333–350 (2022)
13. Li, S., Liu, F., Jiao, L.: Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1395–1403 (2022)
14. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2720–2727 (2013)
15. Lu, Y., Yu, F., Reddy, M.K.K., Wang, Y.: Few-shot scene-adaptive anomaly detection. In: Proceedings of the European Conference on Computer Vision, pp. 125–141 (2020)
16. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 341–349 (2017)
17. Lv, H., Chen, C., Cui, Z., Xu, C., Li, Y., Yang, J.: Learning normal dynamics in videos with meta prototype network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15425–15434 (2021)

18. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1975–1981 (2010)
19. Majhi, S., Dai, R., Kong, Q., Garattoni, L., Francesca, G., Brémond, F.: OE-CTST: outlier-embedded cross temporal scale transformer for weakly-supervised video anomaly detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 8574–8583 (2024)
20. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: a review. *ACM Comput. Surv. (CSUR)* **54**(2), 1–38 (2021)
21. Pourreza, M., Mohammadi, B., Khaki, M., Bouindour, S., Snoussi, H., Sabokrou, M.: G2d: generate to detect anomaly. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2003–2012 (2021)
22. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021)
23. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6479–6488 (2018)
24. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4975–4986 (2021)
25. Wang, Y., et al.: Internvid: a large-scale video-text dataset for multimodal understanding and generation. arXiv preprint [arXiv:2307.06942](https://arxiv.org/abs/2307.06942) (2023)
26. Wu, P., et al.: Not only look, but also listen: learning multimodal violence detection under weak supervision. In: Proceedings of the European Conference on Computer Vision (2020)
27. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision, pp. 305–321 (2018)
28. Zhong, J.X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner: train a plug-and-play action classifier for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1237–1246 (2019)
29. Zhou, H., Yu, J., Yang, W.: Dual memory units with uncertainty regulation for weakly supervised video anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3769–3777 (2023)
30. Zhou, Y., Qu, Y., Xu, X., Shen, F., Song, J., Shen, H.: Batchnorm-based weakly supervised video anomaly detection. arXiv preprint [arXiv:2311.15367](https://arxiv.org/abs/2311.15367) (2023)



# Hypergraph Self-Attention and Channel Topology Specialization Network for Automatic Generation of Labanotation

Weihao Chen<sup>(✉)</sup>, Wanru Xu, and Zhenjiang Miao

Beijing Jiaotong University, Beijing, China  
{22120484,xuwanru,zjmiao}@bjtu.edu.cn

**Abstract.** As more and more attention is paid to the digital protection of traditional culture, Labanotation, as a way to protect traditional culture, has also attracted the attention of scholars. Especially in the field of automatically generating Labanotation, some methods have been proposed. However, existing Labanotation automatic generation methods ignore the high-order kinematic dependencies between the performer's body joints. Furthermore, the static modeling of the channel skeleton topology loses the unique correlation of each channel. Therefore, these methods cannot accurately represent complex dance movements. We propose a Hypergraph Self-Attention (HSA) and Channel Topology Specialization (CTS) network (HSA-CTS) for automatic generation of Labanotation. HSA-CTS includes a global context attention feature extraction module and a local channel topology specialized feature extraction module. First, it models the human body's high-order kinematic dependence on the global spatiotemporal relationship of the skeleton, then dynamically learns the topology in different channels and effectively aggregates joint features in different channels for human action recognition. Experimental results show that our proposed method outperforms state-of-the-art methods on Laban16 and Laban48 (common datasets for Labanotation studies).

**Keywords:** Labanotation · Hypergraph · Self-attention · Channel Topology Specialization

## 1 Introduction

Nowadays, people pay more and more attention to the protection of traditional culture. As a very precious cultural heritage, traditional dance is gradually facing the crisis of being lost due to the lack of effective recording and protection methods. At the beginning of the 20th century, dance theorist Rudolf von Labanotation created Labanotation, a standard notation system for recording dance movements, to unify various types of dances into specifications and record them [1]. This recording system can use specific symbols to express human dance movements simply and effectively, and has since been widely used for the recording and protection of traditional dances. However, manually recording not only

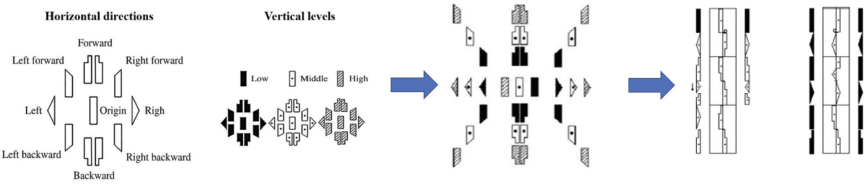
requires professional Labanotation writers, but also consumes a lot of energy from professionals. Therefore, there were some studies on using computers to automatically generate Labanotation scores.

The early traditional method requires pre-segmentation of continuous motion data [2–10], and then modeling and feature extraction of each segment. This method is also called a two-stage method. The main disadvantage of this method is that the quality of data segmentation directly affects subsequent modeling and feature extraction. Some later methods no longer pre-segment the data and use sequence-to-sequence models to align the input skeleton sequence with the output dance spectrum sequence [11–16] to achieve global optimization. These methods are called one-stage methods. The one-stage approach outperforms the two-stage method in both performance and operational complexity. The model is optimized as a whole through sequence-to-sequence alignment. However, these methods ignore the high-order kinematic dependence of human joints. For example, in the long jump movement, the end joints of the arms and the joints of the feet have a synergistic relationship throughout the movement. Traditional graph structures use adjacency matrices based on correlations between joints to represent the topological arrangement of joints. This graph structure is not conducive to extracting the internal relationships between such non-physically connected joints. Secondly, since the skeleton topology modeling of previous methods is channel sharing modeling, information is inevitably lost in multi-channel action recognition tasks. Human actions contains unique information when viewed from different directions.

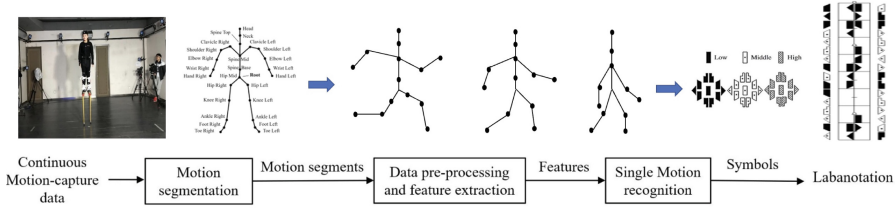
To solve this problem, we propose a Hypergraph Self-Attention and Channel Topology Specialization network for automatic generation of Labanotation. First, the hypergraph [18–20] is an extension of the graph that has good performance in handling human skeletal structures, particularly in tasks involving multi-joint collaboration in the human body. We embed the skeletal hypergraph into a variant of the transformer to connect the global context and capture the spatiotemporal dependencies of higher-order movements in human joints. Additionally, we adopt a channel topology specialization method to model the skeletal topology map. By applying specialized treatment to the topology map of each channel, we achieve finer-grained information capture.

In summary, the contributions of this article are as follows:

- We design a network based on hypergraph self-attention for automatic Labanotation generation. By embedding skeleton information into variants of the transformer, it can better connect with the global context to capture the high-order aspects of human joints Kinematic spatiotemporal dependence.
- We further use a channel topology specialized network to model the skeleton topology map to extract joint features in different channels information and integrate it into the network.
- Our model is evaluated on two datasets, laban16 and laban48, and the results show that our model has achieved the best results so far.



(a) The symbolic description of Labanotation is introduced, which is represented by a combination of horizontal and vertical symbols into specific action symbols, and is finally drawn on the three-line spectrum.



(b) The main process of the two-stage method is introduced, which consists of obtaining motion capture data, segmenting motion segments, processing motion segment data, feature extraction, motion recognition, and Labanotation generation.

**Fig. 1.** The expression of Labanotation symbols is in part (a). A specific action is composed of horizontal and vertical symbols. The main experimental steps of the second phase are in part (b).

## 2 Related Works

Labanotation is one of the most popular standardized dance notation systems, used to record and analyze human movements in dance, splitting a complex overall movement into each basic human movement according to the human body movement parts and spatial divisions. The legend display of Labanotation is shown in Fig. 1 (a).

### 2.1 Traditional Two-Stage Learning Method

In the early stages of research, scholars focused on improving the ability to extract skeleton features and improving human action analysis methods. Guo’s research [4] defines a method that achieves motion segmentation and recognition. In the model proposed by Chen [3], static multi-frame analysis is used to accurately find the locations of human body joints. These methods all segment and re-identify continuous action sequences, which are also called two-stage methods, as shown in Fig. 1 (b). Moreover, they are all methods that focus on spatial modeling and are deficient in temporal modeling.

Subsequent studies focused extensively on capturing temporal relationships between frames in segmented motion. Zhou [5] utilized dynamic time warping

to align motion segments with pre-existing templates, enhancing motion classification. Li [6] introduced an approach to extracting relative joint position features from motion capture data, employing hidden Markov models for temporal analysis. The methodologies proposed by Zhang [7] and Hao [8] incorporated recurrent neural networks (RNNs), showcasing swift and accurate recognition of segmented motions. Additionally, certain investigations [9, 10] enhanced the precision of motion segmentation by analyzing factors such as movement speed and variations in the body’s mass center. Because the methods introduced above require segmentation of continuous motion capture data first, and then feature extraction and recognition, we call them two-stage methods.

## 2.2 Near-Term One-Stage Approach

The main issue with the two-stage method lies in the significant impact of the quality of segmented action data on subsequent recognition tasks. Subsequently, a one-stage method emerged as a viable alternative, eliminating the need for segmented action data. Xie [11] introduced a two-stream fusion-directed graph neural network (DGNN) coupled with connectionist temporal classification (CTC). This approach aimed to enhance recognition performance by leveraging contextual spatiotemporal relationships, emphasizing subtle variations in spatial and temporal attributes. In a similar vein, Li [21] opted for the widely adopted encoder-decoder network to transform the task into a sequence processing problem. However, a drawback of this approach is that the features of the input data are encoded into a fixed-length word vector, resulting in the loss of some spatiotemporal information during the decoding of the output symbol sequence. To address this issue, Gong [16] proposed a dual-stream model, which utilizes a temporal and spatial multi-scale convolutional network (TSMS) to capture short-term spatiotemporal dependencies in the skeleton sequence. Simultaneously, a Transformer network is employed to grasp the long-term temporal and spatial dependencies of the skeleton sequence.

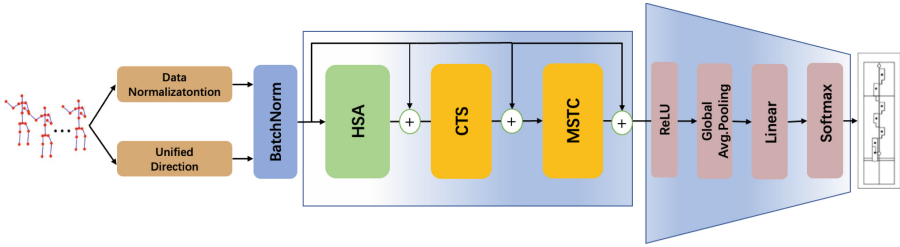
## 2.3 Skeleton Topology Modeling

Researchers have long been exploring better ways to model skeletal topology. For methods that adopt topology-sharing, the approach entails using the same topology for aggregation across all channels, which often leads to the loss of unique features present in different channels. Most existing Graph Convolutional Network (GCN) methods [22–24] employ topology-sharing techniques.

In contrast, non-shared topology methods involve selecting different topologies within different channels. This approach alleviates constraints associated with shared topology, enabling the model to more effectively extract subtle features between different channels. For example, Cheng et al. [27] introduced the Dynamic Channel-GCN (DC-GCN), which assigns individually parameterized topologies to different channel groups. Similarly, the method proposed in this paper employs non-shared topology methods for channel topology modeling.



To the best of my knowledge, this paper presents the first method for generating Labanotation using non-shared topology techniques.



**Fig. 2.** The overall framework of HSA-CTS. The rectangular part is the encoder for feature extraction, and the trapezoidal part is the classifier.

### 3 Proposed Method

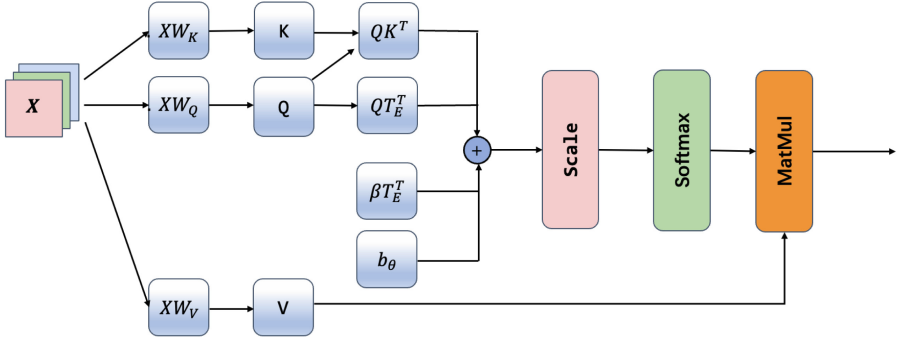
In this part, we will introduce the proposed model framework, specific implementation details, and some mathematical representation methods used. The overall framework is shown in Fig. 2.

First, in order to better capture the intrinsic connections between non-physically connected joints of the human body, we proposed the Hypergraph Self-Attention (HSA) module. The data is input to the Hypergraph Self-Attention module after normalization and direction unification. This module extracts the global contextual high-order dependency information of the skeleton by embedding the skeleton joint information into transformer variants. To extract unique information in each channel, we propose the Channel Topology Specialization (CTS) module. Global information output from HSA is input into CTS. The CTS module uniquely models each channel, extracts unique information in different channels of the skeleton topology, and further extracts local information between skeletal joints, thereby achieving the purpose of refining global information. Finally, the Multi-Scale Temporal Convolution (MSTC) module is used to extract the relationship between different motion data time frames and perform classification output. Below we introduce each part in detail.

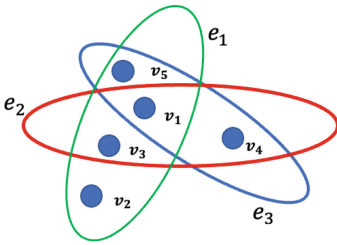
#### 3.1 Hypergraph Self-Attention(HSA) for Joint Feature Extraction

In this part, we will introduce the Hypergraph Self-Attention(HSA) module. The structural implementation is shown in Fig. 3.

Different from the structure and representation of ordinary graphs, edges in hypergraphs are called hyperedges, which can connect two or more vertices. The specific representation is shown on the left side of Fig. 4. An unweighted hypergraph of human skeleton can be defined as  $G_h = (V_h, E_h)$ , where  $V_h$  is the



**Fig. 3.** The internal implementation of HSA. We embed the hyperedge representation and spatial graph distance skeleton structure into the transformer variant to extract skeleton structure features.



	$e_1$	$e_2$	$e_3$
$v_1$	1	1	1
$v_2$	1	0	0
$v_3$	1	1	0
$v_4$	0	1	1
$v_5$	1	0	1

**Fig. 4.** Representation of the relationship between edges and nodes in the hypergraph, and the adjacency matrix.

set of joints, and  $E_h$  is the set of skeleton hyperedges. For hypergraph  $G_h$ , the incidence matrix  $H$  is shown in Fig. 4, which denotes the relationship between nodes. Specifically, if the element in the  $i$ -th row and the  $j$ -th column of the matrix is 1, it means that the  $j$ -th hyperedge contains the  $i$ -th node. Given  $v \in V_h$  and  $e \in E_h$ , the entry of  $H$  is defined as follows:

$$h(v, e) = \begin{cases} 1, & v \in e \\ 0, & v \notin e \end{cases} \tag{1}$$

The degree of a node  $v \in V_h$  represents the number of hyperedge passing through the node, the degree of a hyperedge  $e \in E_h$  represents the number of nodes contained in the hyperedge, their definitions are as follows:

$$d(v) = \sum_{e \in E_h} h(v, e), \quad d(e) = \sum_{v \in V_h} h(v, e) \tag{2}$$

We denote  $D_v$  as the diagonal matrices of node degrees  $d(v)$ ,  $D_e$  as the diagonal matrices of the hyperedge degrees  $d(e)$ , and  $W_e$  as the diagonal matrices of the hyper-edge weights.

Let  $H$  represent the adjacency matrix of the hypergraph, then the normalized adjacency matrix can be expressed as  $\tilde{H} = D_e^{-\frac{1}{2}} H D_e^{-\frac{1}{2}}$ ,  $C$  represents the feature dimension of the input data, The input data  $X \in \mathbb{R}^{V \times C}$  represents joint features, Then the characteristics of the hyperedge  $T_E$  can be regarded as the joint characteristics of the nodes included in the hyperedge:

$$T_E = \tilde{H} X W_e \tag{3}$$

where  $W_e \in \mathbb{R}^{C \times C}$  is the hyperedge weight matrix.

We embed the skeletal graph structure into a variant of the transformer using a method similar to Graphormer [17]. Due to our use of a hypergraph structure, information from hyperedges is embedded separately.

We denote  $X = [X_1^T, \dots, X_n^T]^T \in \mathbb{R}^{n \times c}$  as the input of the self-attention module where  $c$  represents the feature dimension of the input data. The input  $X$  is projected by three matrices  $W_Q \in \mathbb{R}^{c \times d_K}$ ,  $W_K \in \mathbb{R}^{c \times d_K}$ , and  $W_V \in \mathbb{R}^{c \times d_V}$  to the corresponding representations  $Q, K, V$ .

The internal implementation of hypergraph self-attention is shown in Fig. 3. The skeleton feature input after node centrality embedding can be expressed as:

$$\tilde{X}_i = X_i + \sigma_{d(v_i)}, \quad Q_i = \tilde{X}_i W_Q \tag{4}$$

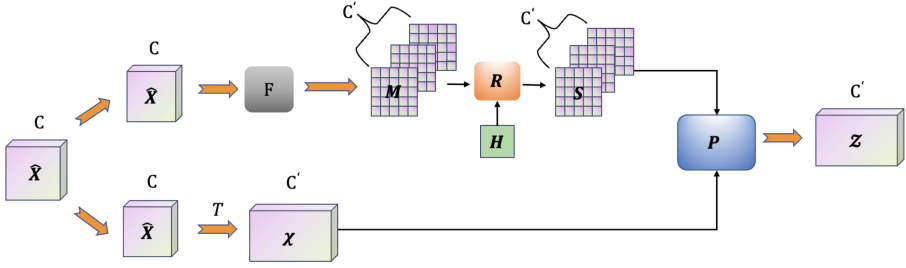
The self-attention of nodes  $v_i$  and  $v_j$  hypergraph after embedding the skeleton graph structural information is expressed as:

$$A_{ij} = \text{softmax}\left(\frac{Q_i K_j^T + Q_i T_E^T + \beta T_E^T}{\sqrt{d_K}} + b_{\theta(v_i, v_j)}\right) V \tag{5}$$

- $Q_i K_j^T$  represents an ordinary attention query between two joints.
- $Q_i T_E^T$  represents an attention query between joints and associated hyperedges.
- $\beta T_E^T$ , aims to calculate the attention bias of different hyperedges independent of the query position, and represents the influence of irrelevant hyperedges on the query position.  $\beta$  represents how much attention bias should be assigned to these irrelevant positions, which is learned during the continuous training of the model.
- $b_{\theta(v_i, v_j)}$  the spatial relation between  $v_i$  and  $v_j$ . The function  $\theta$  can be defined by the connectivity between the nodes in the hypergraph. We choose  $\theta(v_i, v_j)$  to be the distance of the shortest path (SPD) between  $v_i$  and  $v_j$  if the two nodes are connected. If not, we set the output of  $\theta$  to be a special value, -1. We assign each (feasible) output value a learnable scalar which will serve as a bias term in the self-attention module [17].

The output after HSA is expressed as:

$$\hat{X}_i = \sum_{j=1}^n A_{ij} \tag{6}$$



**Fig. 5.** The internal implementation of CTS. The upper side of the module is the unique modeling of the channel, and the lower side is the feature transformation into a high-level feature space.

### 3.2 Channel Topology Specialization(CTS) Network

The overall structure and specific implementation of our proposed CTS module is shown in Fig. 5. We dynamically model the skeleton topology in different channels to extract the correlation between joints in different kinds of motion features, and then aggregate each input feature converted into high-level features through a linear transformation with the corresponding skeleton topology. output. Below we introduce the specific implementation.

**Channel Specialization Modeling.** We have devised a simple modeling function, denoted as  $F$ , to characterize the inter-channel correlations among various key nodes. Let  $\hat{X}_i$  and  $\hat{X}_j$  represent the input features of  $v_i$  and  $v_j$ , respectively, after undergoing HSA. The inter-channel correlation between them can be described as the channel-specific topological relationship resulting from a nonlinear transformation of the distance along a particular channel dimension between  $v_i$  and  $v_j$ . The formula is as follows:

$$F(\hat{X}_i, \hat{X}_j) = \sigma(\hat{X}_i - \hat{X}_j) \tag{7}$$

where  $\sigma(\cdot)$  is activation function and the channel topology relationship  $M \in \mathbb{R}^{n \times n \times \hat{c}}$  between  $v_i$  and  $v_j$  is expressed as:

$$m_{ij} = \gamma(F(\hat{X}_i, \hat{X}_j)) \tag{8}$$

Then use the specialized topology to optimize the globally shared topology to obtain topology modeling with dual information:

$$S = R(H, M) = H + \omega M \tag{9}$$

where  $\omega$  is a learnable parameter,  $H$  is added to each channel of  $\omega M$  for fusion specialization.

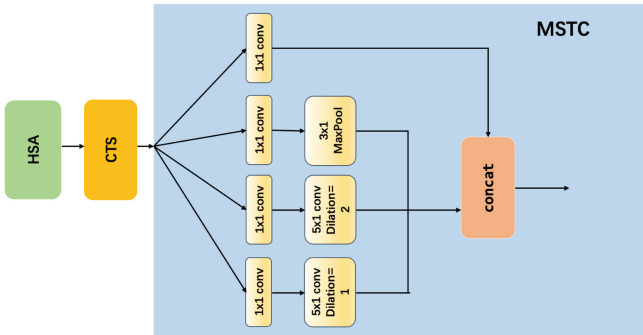
**Channel Feature Transformation.** The purpose of feature transformation is to increase the channel dimension of the input features through  $T(\cdot)$ . The formula is:

$$\mathcal{X} = T(\hat{X}) = \hat{X}W \tag{10}$$

**Feature Aggregation.** By utilizing the aforementioned formula, we have acquired both the specialized channel topology, denoted as  $M$ , and the advanced features, represented by  $\mathcal{X}$ . CTS employs a channel-level strategy for feature aggregation. This entails modeling a distinct channel topology diagram for each channel, where diverse channel topology diagrams encapsulate relationships between nodes corresponding to different motion characteristics. Consequently, feature aggregation is executed on each channel topology map, yielding the final output  $Z$  achieved through the concatenation of output features from all channel maps. This process is formulated as follows:

$$\mathcal{Z} = P(\mathcal{X}, S) = [S_1\mathcal{X}_1 || S_2\mathcal{X}_2 \dots S_{\hat{c}}\mathcal{X}_{\hat{c}}] \tag{11}$$

where  $||$  is concatenate operation.



**Fig. 6.** Internal implementation of multi-scale temporal convolution.

### 3.3 Multi-scale Temporal Convolution(MSTC)

In order to model the temporal correlation of continuous lines of human actions, we adopt the Multi-Scale Temporal Convolution (MSTC) module [24, 25] as the final feature extraction part. This module contains four branches, each containing a  $1 \times 1$  convolution to reduce channel dimension. The first three branches contain two temporal convolutions with different dilations and one Max-Pool respectively following  $1 \times 1$  convolution. The results of the four branches are concatenated to obtain the output. The specific implementation is shown in Fig. 6.

Finally, HSA-CTS is built by alternately stacking HSA, CTS and MSTC layers as follows:

$$Y = ReLU(MSTC(CTS(HSA(X)))) \tag{12}$$

## 4 Experiments

In this part, we introduce the data sets, evaluation indicators, model optimization details in the experiment, and the final actual effect of the model.

### 4.1 Labanotation Datasets

The two datasets we used are Laban16 and Laban48. The number of samples in the two data sets are 1600 and 4800 respectively. Each sample contains 3–8 actions. The number of labels is 7121 and 22654 respectively. The number of frames is 1119706 and 3067156 respectively. The action categories are 16 and 48 respectively, corresponding to 16 and 48 basic Labanotation symbols.

### 4.2 Implementation Details

All experiments are conducted with the PyTorch deep learning library. We train the model using the standard cross entropy loss. The learning rate is initially set to 0.01, with a decay factor of 0.1 applied at epochs [20, 40, 60], respectively. For Laban16 and Laban48, the batch size is set to 32 and 64. The number of heads in multi-head self-attention is set to 9, and the weight decay for Stochastic Gradient Descent (SGD) is set to 0.0005. All experiments are carried out on a Ubuntu server with a 2.2GHz CPU and an NVIDIA Tesla P40 GPU. The CTS part uses three parallel CTS modules to learn the local feature relationships of the human skeleton.

**Table 1.** The results compared with the state-of-the-arts methods on Laban16 and Lanban48.

Model name	Laban16(%)	Laban48(%)
CRNN + CTC [12]	72.80	68.92
Seq2Seq [21]	73.60	70.89
DFGNN-CTC [26]	87.79	82.02
Lie+CRNN + Attention + Seq2seq [13]	86.21	91.61
GS-GCN + RA-Attention + Seq2seq [15]	89.84	89.40
FFCRNN+ Seq2Seq [14]	90.65	93.29
TSMS + SSA + TSA + PSA (SLSTRM) [16]	95.16	95.78
Ours: HSA + MSTC	95.82	96.03
Ours: HSA + CTS + MSTC	<b>95.98</b>	<b>96.22</b>

### 4.3 Results and Analysis

In order to verify the effectiveness of our proposed model HSA-CTS, we conducted ablation experiments on the automatically generated research data sets Laban16 and Laban48 of the Labanotation evaluation study. The effectiveness of the components in HSA-CTS are verified through ablation experiments.

**Comparison to the State-of-the-Arts Methods:** First, we use the Transformer-based HSA global feature extraction network to evaluate the effectiveness of global spatiotemporal features for dance action sequence recognition on Laban16 and Laban48, using the global feature extraction network as our basic comparison reference experiment. The generation results of our proposed network model surpass all previous models with good results, which shows that it is effective for us to embed the hypergraph skeleton topology into the transformer to generate Labanotation.

**Table 2.** Comparison of computational complexity.

Model	Parameters(M)	Laban16(%)	Laban48(%)
SLSTRM [16]	5.8M	95.16	95.78
Ours: HSA+ CTS + MSTC	<b>4.7M</b>	<b>95.98</b>	<b>96.22</b>

**Table 3.** The mean and standard deviation of the model on the Laban16 and Laban48 datasets.

Model	Dataset	Mean Accuracy (%) $\pm$ Standard Deviation
SLSTRM [16]	Laban16/Laban48	95.16/95.78
Ours: HSA + CTS + MSTC	Laban16/lab48	<b>95.912 <math>\pm</math> 0.023/96.174 <math>\pm</math> 0.029</b>

From the results in Table 1, we can see that the HSA we proposed has played a very good role in the task of automatically generating Labanotation. Thanks to the good global context information extraction capability of the transformer model and the good representation of the human skeleton structure by the hypergraph, even if we do not add the CTS module to perform specialized modeling of the skeleton topology to extract local features, the HSA model exhibits outstanding results. After adding the CTS module, the entire model network further strengthens the extraction of local features, and our network effect has been further improved.

We verified the computational complexity of our model and compared it with the SLSTRM [16] method, as shown in Table 2. In addition to the SLSTRM [16] method, we did not compare other indicators such as computational complexity

with other methods because our method has much higher accuracy than theirs. The results show that our method not only has higher accuracy on both datasets, but also has lower computational complexity and better performance. We also conducted multiple experiments to verify the statistical significance of the model, proving that our results are not accidental, as shown in Table 3.

**Table 4.** The effectiveness of each individual component of HSA-CTS.

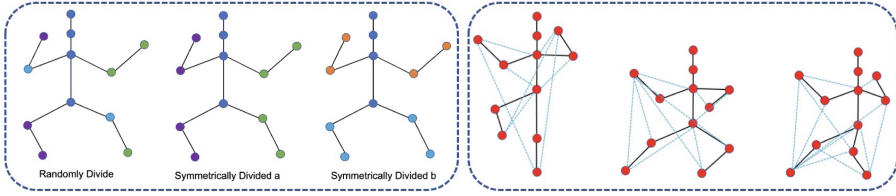
SA	HSA	CTS	MSTC	Laban16(%)	Laban 48(%)
✓				93.21	93.57
	✓			93.98	93.87
✓			✓	94.23	94.54
	✓		✓	95.82	96.03
		✓		93.65	93.13
		✓	✓	95.32	95.22
✓		✓		94.03	94.13
	✓	✓		94.51	95.11
	✓	✓	✓	<b>95.98</b>	<b>96.22</b>

**Ablation Study:** We conducted ablation experiments to validate the effectiveness of each part of the model. In Table 4, we present the comparison results between the hypergraph self-attention module and the ordinary graph self-attention module. From the results, it can be seen that the hypergraph structure performs better in capturing overall information about human joints compared to the ordinary graph structure. This can be attributed to the fact that hyperedges can ignore the physical connections between joints, thereby facilitating feature extraction and information aggregation, significantly improving the upper limit of the model.

In comparison to HSA, CTS focuses more on modeling unique features of the feature channels to capture finer-grained channel characteristics. This was also the original intention behind designing the entire model, as the Labanotation generation task requires consideration of both global and local details. HSA, built upon Transformer, has been experimentally proven to excel in capturing global features, while the inclusion of CTS can compensate for some potentially lost fine-grained features by HSA, thus complementing each other.

However, it is worth noting that both modules extract features in spatial dimensions. MSTC further improves the performance of the entire model by extracting temporal features from the joint motion between frames. It can be seen that the three parts of the entire model They are combined with each other to achieve the best results.





**Fig. 7.** The left side shows the hyperedge division of nodes, and the right side shows the visualization of the change of an action in the dataset. The blue dotted line represents the high-order dependency captured by the hypergraph. (Color figure online)

**Table 5.** Discussion on performance of hyperedge partitioning.

Partition Type	Laban16(%)	Laban 48(%)
Randomly Divide	94.60	95.40
Symmetrically Divided a	95.98	96.22
Symmetrically Divided b	95.31	95.87

### Hypergraph Captures High-Order Dependency Validity Verification:

Our division of hyperedges is fed back into the hypergraph adjacency matrix. The division of hyperedges is crucial to the representation of the skeleton action structure and affects the degree of correlation between joints. We divide the hyperedges based on people’s prior knowledge, as shown in Table 5 and Fig. 7. We found that the best division method is left-right symmetric.

The left side of Fig. 7 shows three types of hyperedge divisions. The joints in the figure do not represent the number of joints in the actual skeleton graph. We use joints of the same color to represent the joint points contained in the same hyperedge. The first one is randomly divided from left to right, the second one divides the human body into two symmetrical left and right parts, and the third one divides the human body into two upper and lower parts. The reason for this division is because we believe that in actual human body movements or data sets, there is an important connection between the joints with the largest difference in motion trajectories. So we visualized the data [28], as shown on the right side of Fig. 7. The blue dotted line represents the high-order dependency captured by the hypergraph.

The experimental results show that the division of hyperedges does have an impact on the recognition accuracy. We analyzed that the second division method achieved the best effect because it divided the ends of the limbs together, and the motion trajectories of these joints were longer.

## 5 Conclusion

In this work, we embed human skeleton structure information into Transformer through a hypergraph structure. This is a better solution for automatic gen-

eration of Labanotation compared to previous models. We propose an HSA structure combined with a CTS module to make the model aware of high-order joint motion dependencies and channel-specific features. The resulting model, called HSA-CTS, achieves state-of-the-art performance. However, the difficulty of obtaining datasets for our task may have some impact on the performance of our proposed model.

## References

1. Guest, A.H.: Labanotation: the System of Analyzing and Recording Movement. Routledge (2013). Recording movement, Psychology Press (2014)
2. Hachimura, K., Nakamura, M.: Method of generating coded description of human body motion from motion-captured data. In: Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication, ROMAN: Cat. No. 01TH8591. vol. 2001, pp. 122–127. IEEE (2001)
3. Chen, H., Qian, G., James, J.: An autonomous dance scoring system using marker-based motion capture. In: 2005 IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4. IEEE (2005)
4. Guo, H., Miao, Z., Zhu, F., Automatic labanotation generation based on human motion capture data. In: Pattern Recognition: 6th Chinese Conference, CCPR, et al.: Changsha, China, 17–19 November 2014, Proceedings, Part I 6, Springer, Berlin, vol. 2014, pp. 426–435 (2014)
5. Zhou Z, Miao Z, Wang J. A system for automatic generation of labanotation from motion capture data. In: 2016 IEEE 13th International Conference on Signal Processing (ICSP), pp. 1031–1034. IEEE (2016)
6. Li, M., Miao, Z., Ma, C.: Dance movement learning for labanotation generation based on motion-captured data. IEEE Access **7**, 161561–161572 (2019)
7. Zhang, X., Miao, Z., Zhang, Q., et al.: Skeleton-based automatic generation of Labanotation with neural networks. J. Electron. Imaging **28**(2), 023026–023026 (2019)
8. Hao S, Miao Z, Wang J, et al. Labanotation generation based on bidirectional gated recurrent units with joint and line features. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4265–4269. IEEE (2019)
9. Wang, J., Miao, Z.: A method of automatically generating Labanotation from human motion capture data. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 854–859. IEEE (2018)
10. Li, M., Miao, Z., Ma, C., et al.: An automatic framework for generating Labanotation scores from continuous motion capture data. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2020)
11. Xie, N., Miao, Z., Wang, J.: Skeleton-based labanotation generation using multi-model aggregation. In: Pattern Recognition: 5th Asian Conference, ACPR, Auckland, New Zealand, 26–29 November 2019, Revised Selected Papers, Part II 5, Springer, vol. 2020, pp. 554–565 (2019)
12. Xie, N., Miao, Z., Wang, J., et al.: End-to-end method for labanotation generation from continuous motion capture data. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2020)
13. Li, M., Miao, Z., Zhang, X.P., et al.: An attention-seq2seq model based on CRNN encoding for automatic labanotation generation from motion capture data. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4185–4189. IEEE (2021)

14. Li, M., Miao, Z., Xu, W.: A CRNN-based attention-seq2seq model with fusion feature for automatic labanotation generation. *Neurocomputing* **454**, 430–440 (2021)
15. Li, M., Miao, Z., Zhang, X.P., et al.: Rhythm-aware sequence-to-sequence learning for labanotation generation with gesture-sensitive graph convolutional encoding. *IEEE Trans. Multimedia* **24**, 1488–1502 (2021)
16. Gong, S., Xu, W., Miao, Z., et al.: Long and short spatial-temporal relations model for automatic generation of Labanotation. *J. Electron. Imaging* **32**(2), 023006–023006 (2023)
17. Ying, C., Cai, T., Luo, S., et al.: Do transformers really perform badly for graph representation? *Adv. Neural. Inf. Process. Syst.* **34**, 28877–28888 (2021)
18. Bai, S., Zhang, F., Torr, P.H.S.: Hypergraph convolution and hypergraph attention. *Pattern Recogn.* **110**, 107637 (2021)
19. Feng, Y., You, H., Zhang, Z., et al.: Hypergraph neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3558–3565 (2019)
20. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: clustering, classification, and embedding. In: *Advances in Neural Information Processing Systems*, vol. 19 (2006)
21. Li, M., Miao, Z., Ma, C.: Sequence-to-sequence labanotation generation based on motion capture data. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4517–4521. IEEE (2020)
22. Plizzari, C., Cannici, M., Matteucci, M.: Spatial temporal transformer network for skeleton-based action recognition. In: *Pattern recognition. ICPR International Workshops and Challenges: Virtual Event, 10–15 January, Proceedings. Part III*, Springer vol. 2021, pp. 694–701 (2021)
23. Huang, Z., Shen, X., Tian, X., et al.: Spatio-temporal inception graph convolutional networks for skeleton-based action recognition. In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2122–2130 (2020)
24. Liu, Z., Zhang, H., Chen, Z., et al.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 143–152 (2020)
25. Chi, H., Ha, M.H., Chi, S., et al.: Infogcn: representation learning for human skeleton-based action recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20186–20196 (2022)
26. Xie, N., Miao, Z., Zhang, X.P., et al.: Sequential gesture learning for continuous labanotation generation based on the fusion of graph neural networks. *IEEE Trans. Circuits Syst. Video Technol.* **32**(6), 3722–3734 (2021)
27. Cheng, K., Zhang, Y., Cao, C., et al.: Decoupling gcN with dropgraph module for skeleton-based action recognition. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XXIV 16*, Springer, pp. 536–553 (2020)
28. Hao, X., Li, J., Guo, Y., et al.: Hypergraph neural network for skeleton-based action recognitio. *IEEE Trans. Image Process.* **30**, 2263–2275 (2021)



# JS-Siamese: Generalized Zero Shot Learning for IMU-based Human Activity Recognition

Mohammad Al-Saad<sup>1,2</sup>(✉), Lakshmish Ramaswamy<sup>1</sup>,  
and Suchendra M. Bhandarkar<sup>1</sup>

<sup>1</sup> School of Computing, University of Georgia, Athens, GA 30602, USA  
mohammad.alsaad@uga.edu

<sup>2</sup> CISE Department, University of Florida, Gainesville, FL 32606, USA

**Abstract.** Inertial Measurement Unit (IMU)-based Human Activity Recognition (HAR) systems that employ Generalized Zero-Shot Learning (GZSL) face significant challenges in accurately classifying activities that were not observed previously during training. These challenges stem primarily from the inherent difficulty of recognizing unseen classes without sacrificing the classification accuracy of observed classes in a GZSL setting. A novel deep neural network (DNN) architecture termed as the *Joint Sequences (JS)-Siamese* architecture is proposed to address these challenges using IMU and video data. The proposed architecture uses skeleton joint sequences to bridge the gap between IMU features and video data, thus effectively solving the domain shift problem. A Siamese DNN-based metric learning model is employed to handle the *hubness problem* by mapping similar samples in close proximity and dissimilar ones farther apart in a joint embedding space. Additionally, a Dynamic Calibration Ensemble (DCE) technique is introduced to address the classification bias towards the observed classes in GZSL, thereby ensuring balanced representation of both, observed and unseen classes. The proposed JS-Siamese DNN architecture is shown to yield significant performance improvement over attribute-based, word embedding-based and video embedding-based GZSL approaches for HAR proposed in the literature. Experimental evaluation on three IMU benchmark datasets, i.e., PAMAP2, DaLiAc and UTD-MHAD demonstrate the effectiveness of the proposed JS-Siamese DNN architecture for sensor-based HAR.

**Keywords:** human activity recognition · generalized zero-shot learning · inertial measurement unit · deep neural network

## 1 Introduction

Human activity recognition (HAR) systems focus on the automated classification of human activities and represent an emerging area of study in mobile and ubiquitous computing. HAR systems often struggle to adapt to complex

situations where human activities not previously observed during training are encountered during testing. In such scenarios, the HAR system is prone to inaccuracies, often mislabeling previously unseen activities as one of the previously observed categories it has been trained to recognize. A fundamental challenge in HAR is dealing with the vast diversity of human activities in contrast to the limited range included in existing benchmark training datasets. Most Inertial Measurement Unit (IMU) training datasets contain fewer than 20 activity labels [15, 16], suggesting a high likelihood of users performing unrecorded activities. While expanding IMU datasets to include a broader range of activities is feasible, the substantial cost associated with data collection and annotation is a major deterrent.

Zero Shot Learning (ZSL) aims to develop a model capable of identifying unseen classes via transfer of knowledge from previously observed classes to unseen classes by embedding both observed and unseen classes in a high-dimensional semantic vector space. These semantic representations can take various forms, such as context-based embeddings, manually crafted attribute vectors [9] automated word vector retrieval, or a combination of the above. Thus, ZSL uses semantic data to bridge the divide between the observed and unseen classes. Traditional ZSL methods, which only include samples from unseen classes in their test sets, do not fully represent real-world scenarios. In reality, samples from observed classes are more common than those from unseen ones, making it impractical to classify only the unseen samples at test time. A more practical and realistic extension of ZSL, termed as Generalized Zero-Shot Learning (GZSL) [3], entails identifying samples from both observed and unseen classes at test time.

## 2 Related Work

Early works on ZSL in the context of IMU-based HAR employ primarily attribute-based approach. The seminal work on *Direct Attribute Prediction* (DAP) [9] implements several Support Vector Machine (SVM) classifiers to separately map and predict the binary attributes in a semantic space. Each semantic attribute is classified by a single SVM classifier followed by the computation of a maximum *a posteriori* (MAP) estimate to derive the final predictions. Inspired by the DAP approach, Cheng et al. [6] propose a method that predicts each binary attribute using a separate SVM classifier in a binary attribute semantic space, followed by  $k$ -nearest-neighbor ( $k$ -NN) classification to yield the final prediction. In a further extension, Cheng et al. [5] implement a conditional random field (CRF) to predict each attribute followed by a  $k$ -NN classifier enhanced with a junction tree algorithm for final classification. Wang et al. [20] present a nonlinear compatibility model to compute compatibility scores in a semantic attribute space between the feature space instances from sensor readings and prototypes from each class. Wu et al. [21] employ a neural network to project the feature instances into a semantic space and perform classification by computing the similarities between the feature instances and the prototypes.

Ohashi et al. [13] leveraged a *Convolutional Neural Network* (CNN) to extract features from raw IMU data while simultaneously performing projection. The attributes are manually assigned weights to denote their importance and stored in an importance table. There are several limitations associated with attribute-based approaches, the most significant being the need for expert knowledge to define attributes. This results in attribute variations caused by human subjectivity and lack of scalability with increasing number of classes. Furthermore, the performance of attribute-based approaches depends heavily on class-specific attribute differences, making it challenging to define attributes for all possible classes.

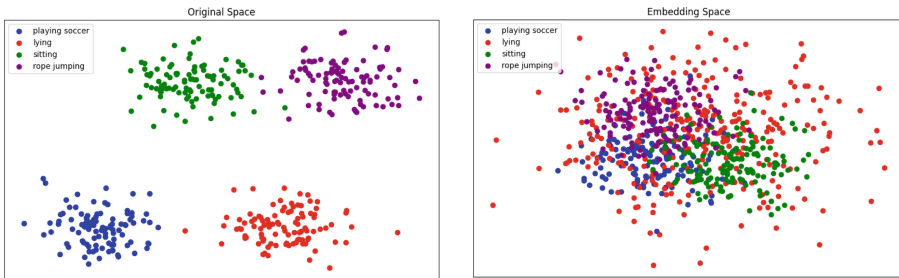
To overcome the limitations of attribute-based approaches, some studies employ a word embedding-based semantic space where the word embeddings are generated using unsupervised learning algorithms on general text corpora such as Wikipedia. The semantic space is generated using embedding vectors that represent words associated with each unknown activity class. Word embedding-based approaches have been shown to be more scalable than their attribute-based counterparts as new classes can be added by simply adding new words. Matsuki et al. [12] compare the embeddings of similar words to hand-crafted attribute vectors and found the recognition performance to be similar. Wu et al. [21] propose a Multi-layer Perceptron (MLP) model with skip connections for projection of word embedding and attribute vectors and experimentally demonstrate that the attribute-based semantic space outperforms the word embedding-based semantic space in terms of classification accuracy. Although the word embedding-based semantic space method is scalable, it suffers from representation complexity, meaning ambiguity and performance instability since the HAR accuracy is observed to be greatly impacted by the learning task at hand and the text corpus used.

Since words lack motion-specific information, Tong et al. [18] propose a video-based semantic space for ZSL in the context of ZSL for HAR. The video semantic space facilitates the transfer of knowledge from a pretrained video action recognition model to unseen activity recognition. While their research shows that a computer vision-derived semantic space has a great potential for improving IMU-based HAR, the methodology described in their paper suffers from the hubness, bias, and domain shift problems that have been addressed in this paper.

In summary, the few studies exploring ZSL for IMU-based HAR utilize primarily a manually crafted attribute vector space, a commonly used semantic space for joint embedding in ZSL applications [6, 12, 20, 21]. Semantic space formulations derived from word embeddings of class labels or class descriptions have shown varying degrees of effectiveness compared to manually crafted attributes [12, 21]. To our knowledge, the work presented by Tong et al. [18] is the only one that employs a semantic space using video embeddings from pretrained video-based HAR models albeit with the associated hubness, bias and domain shift problems which are addressed in this paper.

### 3 Contributions

In light of the related work, this paper addresses some key challenges in GZSL for sensor-based HAR. The first is the domain gap between the IMU features and video embeddings, which results in the *domain shift* problem arising from the mismatch or discrepancy in data distributions between the training phase and the testing phase. The domain shift can adversely impact the model’s generalizability and consequently the accuracy of its predictions. Second, the study utilizes a projection-based technique to map instances from the IMU feature space into the video semantic space followed by nearest neighbor classification to determine the predicted class. However, this approach is hindered by the *hubness* problem, a well-documented issue in the context of zero-shot learning (ZSL) and generalized zero-shot learning (GZSL). Hubness occurs when certain data points, referred to as “hubs” become overly central within the high-dimensional feature space. As shown in Figure 1, these hub points frequently appear as the nearest neighbors for many other data points, irrespective of their actual class similarity. This phenomenon is a facet of the curse of dimensionality, significantly impacting the effectiveness of nearest neighbor methods. In high-dimensional spaces, the distance metrics used to determine similarity become less discriminative. This cause some points to be considered close to many others. These hubs distort the neighborhood structure, leading to incorrect nearest neighbor classifications. Consequently, the hubness problem complicates the crucial GZSL task of accurately computing similarity or distance measures between data points. In the context of this study, this results in certain IMU features instances being projected too frequently into the video embedding space as similar to multiple other instances, thereby degrading the classification accuracy.



**Fig. 1.** Hubness Problem in GZSL

Third, since the GZSL-based classifier is expected to encounter both familiar (i.e., previously observed) and unfamiliar (i.e., unseen) classes during test time in real-world scenarios, the classifier tends to exhibit a strong *bias* towards the previously observed (i.e., seen) classes, leading to frequent misclassification of the unseen classes.

In this paper a novel and effective deep neural network (DNN) architecture termed as the *Joint Sequences (JS)-Siamese* architecture is proposed to address the aforementioned challenges of GZSL in the context of IMU-based HAR. Specifically, the paper makes the following contributions:

- It aims to overcome the domain shift problem by employing a common data modality comprising of skeleton joint sequences. The underlying idea is that skeleton key points or joints effectively encapsulate the movements of body parts in a video. Thus, models that use joint sequence data focus on the motion-related information content in the input videos, in a manner similar to IMU data. The domain shift problem is addressed by systematically integrating a set of existing techniques to convert IMU data to a skeleton-like representation while simultaneously extracting skeleton joint sequences from input video data using a skeleton HAR model. Thus, the proposed approach matches the data distribution from both modalities thereby addressing the domain shift problem in the context of GZSL.
- It addresses the hubness problem by employing a Siamese DNN that incorporates deep metric learning. Hub points in the feature space are avoided by mapping similar samples in close proximity and dissimilar ones farther apart in a lower-dimensional shared embedding space.
- A novel *Dynamic Calibration Ensemble* (DCE) approach is proposed to address the inherent bias problem in the context of GZSL. The proposed DCE approach aims to balance the representation of both seen and unseen classes via a dynamic calibration mechanism that adjusts the biases of individual classifiers over time, based on variations in their performance.

The effectiveness of the proposed JS-Siamese DNN architecture is demonstrated by performing extensive experiments on three IMU benchmark datasets including PAMAP2 [15], DaLiAc [10] and UTD-MHAD [4].

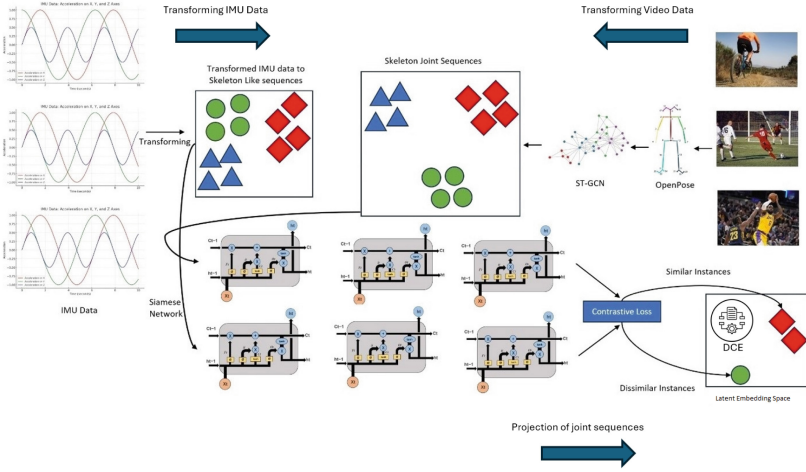
## 4 JS-Siamese DNN Architecture

The proposed JS-Siamese architecture, depicted in Figure 2, is comprised of four main components: the first is the transformation of the IMU data into skeleton joint sequences using a combination of existing techniques; the second is the extraction of joint sequences from input video data using the *Spatial Temporal Graph Convolutional Network* (ST-GCN) model [22]; the third is the projection of joint sequences derived from the IMU and video data into a shared embedding space; and the fourth is the proposed DCE technique used to address the bias problem in classification for HAR.

### 4.1 Transformation of IMU data into skeleton joint sequences

The IMU sensor data, comprising of accelerometer and gyroscope measurements, is preprocessed and transformed into a skeleton joint sequences representation using a novel approach that systematically leverages and integrates a set of





**Fig. 2.** The proposed JS-Siemese DNN Architecture

existing techniques. The accelerometer data is normalized using the standard acceleration due to gravity and subsequently fused with the gyroscope data using the Madgwick filter [11] to estimate the orientation of the sensor. The Madgwick filter computes a quaternion representing the orientation of the sensor relative to the Earth. The Madgwick filter comprises of a prediction step using the gyroscope data and an update step that entails a correction using the accelerometer data to ensure that the sensor orientation aligns with the direction of gravity. The new quaternion is then normalized after each update to ensure that it represents a valid rotation. The orientation quaternion computed by the Madgwick filter is then used to transform the accelerometer data. This transformation allows one to use the orientation data to rotate the accelerometer vector such that the accelerometer vector is now represented in a consistent frame of reference throughout the dataset, which helps to alleviate the effects of sensor drift over time.

The orientations of the joints are estimated from the fused sensor data using Euler angles [14]. Euler angles provide a straightforward and intuitive representation of joint orientations by means of rotations around the three axes (i.e., roll, pitch, and yaw). We compute the rotation matrix from the quaternion and subsequently extract the Euler angles. To obtain a skeleton-like sequence representation, the joint coordinates are aggregated over time into a fixed-length representation for each activity instance using attention-based temporal pooling. Given a matrix  $C$  representing the joint coordinates, where each row  $i$  corresponds to a time step and each column  $j$  corresponds to a specific joint coordinate, a pooled representation is derived across the time step. The attention weights are computed based on the absolute values of joint coordinates as  $W_{i,j} = |C_{i,j}|$  where  $W$  is the attention weights matrix with the same dimensions

as matrix  $C$ . The attention scores for each time step  $i$  are derived by averaging the attention weights across all joint coordinates as  $S_i = \frac{1}{J} \sum_{j=1}^J W_{i,j}$  where  $S$  is a vector of attention scores and  $J$  represents the total number of joint coordinates. To obtain the pooled representation of joint coordinates, a weighted average is computed across all time steps using the attention scores as follows:

$$P_j = \frac{\sum_{i=1}^I S_i \cdot C_{i,j}}{\sum_{i=1}^I S_i} \quad (1)$$

where  $P$  represents the pooled representation,  $I$  is the total number of time steps and  $j$  is the joint coordinate under consideration.

## 4.2 Extraction of joint sequences from video data

The joint sequences are extracted from video data using a pre-trained ST-GCN model [22] resulting in motion-level information which is considered critical for HAR. The OpenPose library [2] is used to detect and extract the human skeleton in each frame of the video and create a mapping that correlates human anatomical joint names (e.g., *left wrist*) with their corresponding numerical indices. These numerical indices are important for locating the spatial coordinates of the joints within the output representation generated by the model. The skeleton data is then fed to the ST-GCN model to generate the joint coordinates from the input video data. To extract the joint-level features from the input video data, the last layer of ST-GCN model is discarded and the output of the penultimate ST-GCN layer deemed to be the joint features representing the coordinates or features of the respective joints in the video frame. The joint coordinates are accumulated over all the video frames to obtain the joint sequences. The result is a set of joint sequences, each sequence encapsulating the positional information of a joint over time.

## 4.3 Projection of joint sequences data

In the proposed HAR scheme, the transformed skeleton-like representations derived from IMU data are one input data modality whereas the joint sequences generated from the video data using the ST-GCN are the other input data modality. A deep metric learning model based on a Siamese DNN is employed to map samples from both modalities in a shared embedding space in a manner such that similar samples from both modalities are mapped in close spatial proximity whereas dissimilar samples are mapped farther apart in the shared embedding space in which the final classification is subsequently performed. Each subnetwork within the Siamese DNN consists of three long short-term memory (LSTM) layers for processing of the joint sequences. The final layer in each subnetwork serves as a projection layer that transforms the final LSTM hidden state into a latent representation and projects it into a lower-dimensional common latent space spanned by vectors of size 64.

Formally, the LSTM processes an input sequence  $x = \{x_1, x_2, \dots, x_T\}$  of length  $T$  and produces a sequence of hidden states  $h = \{h_1, h_2, \dots, h_T\}$  using  $h_t = LSTM(x_t, h_{t-1})$  where  $h_t$  is the hidden state at time  $t$ . The last hidden state  $h_T$  is extracted and processed by the projection layer resulting in a vector  $z = W \cdot h_T + b$  where  $W$  and  $b$  are the weight matrix and bias respectively. We employ a contrastive loss function based on cosine similarity [8]. Given a pair of sequences  $x_1$  and  $x_2$  and their latent representations  $z_1$  and  $z_2$  generated by the Siamese network, the cosine similarity  $CS$  between  $z_1$  and  $z_2$  is computed as:

$$CS(z_1, z_2) = \frac{z_1 \cdot z_2}{\|z_1\|_2 \cdot \|z_2\|_2} \quad (2)$$

The contrastive loss  $L$  for  $z_1$  and  $z_2$  is given by:

$$L(z_1, z_2) = \begin{cases} CS(z_1, z_2)^2 & \text{if } label = 0 \\ \max(0, \delta - CS(z_1, z_2))^2 & \text{if } label = 1 \end{cases} \quad (3)$$

where  $label = 0$  denotes similar sequences,  $label = 1$  denotes dissimilar sequences and the margin  $\delta$  is a hyperparameter that specifies the minimum separation distance between dissimilar sequences in the cosine similarity space. For similar sequences  $z_1$  and  $z_2$ , the contrastive loss  $L(z_1, z_2)$  is minimized when  $CS(z_1, z_2) \approx 0$  whereas for dissimilar sequences  $z_1$  and  $z_2$ , the contrastive loss  $L(z_1, z_2)$  is minimized when  $CS(z_1, z_2) \leq \delta$ . To train the Siamese network, we train the model for 50 epochs using an Adam optimizer with a learning rate of 0.001.

#### 4.4 Dynamic calibration ensemble (DCE)

*Calibrated Stacking* [3] is one of the key approaches used to address the bias problem in GZSL. The main idea behind calibrated stacking is to adjust the scores of the *seen* class discriminant function using a calibration factor. The calibrated stacking rule is follows:

$$\hat{y} = \arg \max_{c \in \mathcal{T}} f_c(x) - \gamma \mathbb{I}[c \in \mathcal{S}] \quad (4)$$

where  $\mathcal{T} = \mathcal{S} \cup \mathcal{U}$  is the union of the set of seen classes  $\mathcal{S}$  and set of unseen classes  $\mathcal{U}$ ,  $f_c(x)$  is a discriminant scoring function for class  $c \in \mathcal{T}$ ,  $\hat{y}$  is the derived class label for input  $x$ ,  $\mathbb{I}[\cdot] \in \{0, 1\}$  indicates whether  $c$  is a seen class or not and  $\gamma$  is a calibration factor. The value of  $\gamma$  determines the best balance between seen and unseen predictions. Calibrated stacking suffers from some significant limitations. First, the calibration hyperparameter used to determine the optimal value requires careful tuning which is time-consuming and task- or dataset-dependent. Second, calibrated stacking may not generalize well across different datasets, i.e., calibration that works well for one dataset may not be suitable for another. Third, calibrated stacking aims to balance the performance between seen and unseen classes. It often involves a trade-off in that improving performance on

unseen classes may result in a decrease in classification accuracy for seen classes, and vice versa.

To address these limitations, we propose a novel *Dynamic Calibration Ensemble* (DCE) scheme that aims to provide a balanced and fair representation of both seen and unseen classes in GZSL. DCE employs an ensemble of stacked classifiers, each fine-tuned to have a specific inherent bias, towards either seen or unseen classes. The novelty of the DCE scheme lies in its dynamic calibration mechanism, which dynamically adjusts the bias of each classifier in the ensemble over multiple epochs in response to observed performance disparities. Thus, the DCE approach adapts to the changing needs of the GZSL task, ensuring that neither the seen nor the unseen classes are disproportionately favored. This mitigates the common issue of bias towards the seen classes in GZSL, resulting in improved generalization performance of the classifier.

The DCE consists of four sets of stacked classifiers, each with its own calibration factor  $\gamma_i$ . The stacked classifiers are designed to have varying degrees of bias with the first set being *strongly biased* towards the *seen* classes, the second set *slightly biased* towards the *seen* classes, the third set *slightly biased* towards the *unseen* classes, and the fourth set *strongly biased* towards *unseen*. This leads to the following change in the calibrated stacking formula:

$$\delta_i(e) = \arg \max_{c \in \mathcal{T}} f_c(x) - \gamma_i(e) || [c \in \mathcal{S}] \quad (5)$$

Each set of stacked classifiers is initially calibrated with a base calibration factor  $\gamma_{base}$ , which sets its inherent bias. These base calibration factors are dynamically adjusted based on the observed disparity in performance (i.e., classification accuracy  $A$ ) with increasing values of the epoch number  $e$ . The disparity  $H$  (i.e., H-score) is computed as the harmonic mean between seen class performance  $A_{\mathcal{S}}$  and unseen class performance  $A_{\mathcal{U}}$  as follows:

$$H = \frac{2 \times A_{\mathcal{S}} \times A_{\mathcal{U}}}{A_{\mathcal{S}} + A_{\mathcal{U}}} \quad (6)$$

The harmonic mean  $H$  (H-score) penalizes the model if either  $A_{\mathcal{S}}$  or  $A_{\mathcal{U}}$  is low, thus ensuring a balanced performance.

The dynamic calibration for the first set of stacked classifiers is computed as  $\gamma_1(e) = \gamma_{base} + \alpha \times \tanh(H)$  where  $\gamma_{base}$  is a base calibration value that ensures an inherent bias in the network and  $e$  is the epoch number. This value is typically set to maintain the model's strong inclination towards seen classes. For this set of stacked classifier, we set  $\gamma_{base} = -10$ . The parameter  $\alpha$  is a scaling factor which determines the strength of the adjustment based on the disparity  $H$  and  $\tanh(H)$  is the hyperbolic tangent of the disparity. The  $\tanh$  function maps any input value to a value between  $-1$  and  $1$ . As the disparity  $H$  grows, this term becomes more influential, adjusting the calibration factor accordingly. For  $\delta_1$ , equation (5) implies that as the disparity increases (indicating that the model is not performing well on unseen classes), the calibration factor  $\gamma_1$  increases, reducing slightly its strong bias towards seen classes.

For the second set of stacked classifiers, the dynamic calibration is defined as  $\gamma_2(e) = \gamma_{base} + 0.5 \times \alpha \times \tanh(H)$ . For  $\delta_2$ , which slightly favors seen classes, the introduction of the multiplier 0.5 indicates that this set adjusts its calibration at half the rate of  $\delta_1$ . This more modest adjustment rate reflects  $\delta_2$ 's initial slight bias towards seen classes. For this set, we initialize  $\gamma_{base} = -5$ . The dynamic calibration for the third set of stacked classifiers is computed as  $\gamma_3(e) = \gamma_{base} - 0.5 \times \alpha \times \tanh(H)$ . For  $\delta_3$ , which slightly favors unseen classes, the calibration factor decreases as disparity increases. As the model underperforms on unseen classes,  $\delta_3$ 's slight increase towards unseen classes is accentuated. For this set, we initialize  $\gamma_{base} = 5$ . Finally, for the fourth set of stacked classifiers, the dynamic calibration is computed as  $\gamma_4(e) = \gamma_{base} - \alpha \times \tanh(H)$ . For  $\delta_4$ , equation (5) implies that the set of stacked classifiers strongly favors unseen classes. Here, we initialize the inherent bias  $\gamma_{base} = 10$ . We followed the work in [19] which tested values of  $\gamma$  ranging from [0.5, 5]. Based on this approach, we initialized the  $\gamma_{base}$  parameters within the range [-10, 10] to ensure a broader coverage for addressing biases towards both seen and unseen classes. The choice of  $\gamma_{base}$  impacts the initial bias of the classifiers, which in turn influences the dynamic calibration process. Different initial values can lead to variations in the convergence behavior of the model and its final performance metrics. If  $\gamma_{base}$  is too low (a large negative value) or set too high (a large positive value), it can cause the model to be overly biased towards either seen or unseen classes, leading to suboptimal performance. While the initial values of  $\gamma_{base}$  play a role in the model's bias and convergence, the DCE mechanism mitigates the impact of these initializations by dynamically adjusting the biases during training. It is recommended to choose initial values within the ranges that provide a reasonable starting point such as [-10,10], as extreme values can affect the speed and efficiency of the dynamic calibration process.

Computing the optimal weights for each stack in the DCE is essential for maximizing performance. The goal is to assign weights that show each stack's ability to balance the recognition of seen and unseen classes. Since the harmonic mean score (H-score) is used to track the performance balance for each stack of classifiers, we use the H-score values directly to calculate the weights as follows:

$$w_i = \frac{H_i}{\sum_{j=1}^N H_j} \quad (7)$$

where  $w_i$  and  $H_i$  are the weight and H-score respectively for the  $i^{th}$  classifier and  $N$  is the total number of classifiers in the set of stacked classifiers. The final prediction is computed as follows:

$$y_{pred} = \arg \max_{c \in T} \left( \sum_{i=1}^4 w_i \times \delta_i \right) \quad (8)$$

## 5 Experimental Results

### 5.1 Data

**IMU data.** Three sensor-based benchmark datasets were used for evaluation of the proposed approach: PAMAP2 [15], DaLiAc [10], and UTD-MHAD [4]. For each dataset, we use the triaxial accelerometer data and gyroscope data in our experiments. For performance assessment, we divide the activity classes within each dataset into *seen* and *unseen* classes. We also categorize the activities in each dataset adopting the activity types defined in [18]. Our approach involves selecting one class from each activity type as *unseen*, thus guaranteeing a balanced representation of different activity types and allowing us to assess the proposed method’s performance across diverse activity types.

In the PAMAP2 dataset, we designate activities such as *playing soccer*, *car driving*, *sitting*, *folding laundry* and *descending stairs* as *unseen*. The *unseen* class instances form part of the testing data, while instances from *seen* classes, such as *watching TV* (primarily subject 101) and *others* (subject 108), comprise the rest of the testing data. The remaining data serves as training instances. In total, we have 13 *seen* classes, 5 *unseen* classes, 18398 training instances and 8300 testing instances.

In the DaLiAc dataset, *unseen* classes include *rope jumping*, *vacuuming*, *standing* and *descending stairs*. Instances from the aforementioned *unseen* classes are used as part of the testing data. Furthermore, within the *seen* classes, we choose examples from subjects 17, 18, and 19 as testing data and use the remainder as training data. In total, we have 8 *seen* classes, 4 *unseen* classes, 15970 training instances and 6108 testing instances.

In the UTD-MHAD dataset, classes such as *draw x*, *bowling*, *swipe left*, *clap*, and *basketball shoot*, are marked as *unseen*. Given the uniform distribution of activity instances across classes in this dataset, the activities are picked randomly from different activity types. Instances from the aforementioned *unseen* classes are included as part of the testing data, whereas for *seen* classes, we select data from subject 8 for testing and use the rest for training. In total, we have 22 *seen* classes, 5 *unseen* classes, 615 training instances and 246 testing instances.

**Video data.** We follow [18] in selecting 10 skeletal video clips per activity class for the 18 activities in the PAMAP2 dataset, and 13 activities in the DaLiAc dataset ensuring at least one skeletal video clip for each *seen* and *unseen* activity. We also collect video data from publicly available datasets such as Kinetics-400[7] and UCF101[17]. We randomly choose 512 consecutive frames ( $\approx 17$  seconds) from each video clip (and repeating shorter video clips to fill 512 frames) as input to the ST-GCN model. We extract the skeleton joint sequences for UTD-MHAD from the videos supplied by the dataset. For each class label, there are a total of 32 video clips available as each activity is performed four times by eight subjects. Since the subjects are always in the center of the frame and the background and camera are fixed, the videos have little variability. For consistency, we utilize 256 consecutive frames to extract the skeleton joint sequences using the ST-GCN model.

## 5.2 Evaluation metrics

In the GZSL setting, two primary metrics are employed for evaluating the proposed approach. First, to assess the accuracy of recognizing activities from both seen and unseen classes, we calculate the average per-class accuracy for seen and unseen classes separately. The average per-class accuracy  $A$  is determined as the mean of the accuracy values for each class, formulated as follows:

$$A = \frac{1}{N_C} \sum_{i=1}^{N_c} \frac{\text{number of correctly classified instances for class } c_i}{\text{number of instances in class } c_i} \quad (9)$$

where  $N_C$  represents the number of classes in either the seen or unseen sets. This metric remains unaffected by class imbalance, providing a balanced evaluation across classes. We denote the average per-class accuracy for seen classes as  $A_S$ , and for unseen classes as  $A_U$ . Subsequently, we compute the H-score as the harmonic mean of these accuracies as shown in equation (6) to provide a single metric that captures the performance across both seen and unseen classes. In our analysis, we thoroughly evaluate the metrics  $A_S$ ,  $A_U$  and H-score. Our primary focus is on the H-score, as it integrates the performance on both seen and unseen classes, offering a comprehensive evaluation of the proposed approach.

## 5.3 Comparison with baselines

We implemented two versions of the proposed JS-Siamese architecture, one with conventional calibrated stacking (JS-Siamese-c) and the other with DCE (JS-Siamese-DCE). We compared both JS-Siamese versions with state-of-the-art (SOTA) methods for sensor-based HAR that employ GZSL. To ensure a fair comparison, we incorporated calibrated stacking in each of the SOTA methods to address the inherent bias problem in GZSL. We use the same train-test split discussed in Section 5.1 for evaluating all the methods. The results of the evaluation are summarized in Table 1.

We compare the JS-Siamese-c architecture to SOTA semantic attribute-based approaches with calibrated stacking i.e., NuActiv-c, CRF+NN-c, NCBM-c, HDPoseDS-c, and EmCoGM-c. The JS-Siamese-c architecture is observed to yield higher classification accuracy values on both *seen* classes ( $A_S$ ) and *unseen* classes ( $A_U$ ) and higher H-score values across all the three benchmark datasets, i.e., PAMAP2, DaLiAc and UTD-MHAD. Specifically, the JS-Siamese-c architecture shows a 16.4% increase in  $A_S$ , 12.3% increase in  $A_U$ , and 15.1% increase in H-score values compared to the best performing SOTA semantic attribute-based approach (in terms of the specific performance measure) when averaged across all the three benchmark datasets.

When comparing JS-Siamese-c to semantic word embedding-based approaches with calibrated stacking such as ExpWord-c and MLCLM-c, JS-Siamese-c is observed to exhibit superior performance in terms of  $A_S$ ,  $A_U$  and H-score values. Specifically, JS-Siamese-c shows a 23.2% increase in  $A_S$ ,

**Table 1.** Comparison of the JS-Siamese architecture to SOTA methods

Approach	Space	PAMAP2			DaLiAc			UTD-MHAD		
		$A_S$ (%)	$A_U$ (%)	$H$	$A_S$ (%)	$A_U$ (%)	$H$	$A_S$ (%)	$A_U$ (%)	$H$
NuActiv-c	Attribute	26.4	27.5	26.9	63.1	33.2	43.5	29.5	23.4	26.1
CRF+NN-c	Attribute	31.6	29.6	30.5	71.3	40.2	51.4	35.2	23.3	28.0
NCBM-c	Attribute	53.6	34.6	42.1	70.0	41.6	52.1	26.1	51.5	34.6
HDPoseDS-c	Attribute	58.3	22.3	32.2	73.2	37.8	49.8	28.2	29.6	28.8
EmCoGM-c	Attribute	57.2	40.6	47.5	58.0	51.6	54.6	39.3	36.8	38.0
ExpWord-c	Word	48.8	39.1	43.3	54.1	44.2	48.6	25.2	26.5	25.8
MLCLM-c	Word	42.0	32.5	36.7	49.8	41.5	45.2	20.0	33.0	24.9
I3D-emb-c	RGB Videos	49.3	41.5	45.0	53.7	44.7	48.7	36.7	34.9	35.7
JS-Siamese-c	Skeletal Videos	63.4	49.1	55.3	77.8	56.7	65.5	48.4	39.8	43.6
JS-Siamese-DCE	Skeletal Videos	69.8	61.2	<b>65.2</b>	85.7	67.5	<b>77.3</b>	51.4	47.3	<b>49.2</b>

12.4% increase in  $A_U$ , and 17.3% increase in H-score values compared to the best performing SOTA semantic word embedding-based approach (in terms of the specific performance measure) when averaged across all the three benchmark datasets. Word embedding-based approaches often face challenges arising from meaning ambiguity and representation complexity resulting in less stable semantic spaces. Also, mapping the IMU-derived activities to words is extremely challenging due to the large domain gap between IMU data and words. However, since JS-Siamese-c uses skeletal joint sequences, it employs a more stable and representative semantic space resulting in enhanced performance in recognizing a diverse range of activities over all the benchmark datasets. Likewise, when compared with video embedding-based approaches such as I3D-emb-c, JS-Siamese-c demonstrated a clear superiority in performance over all three benchmark datasets. In particular, JS-Siamese-c shows a 16.3% increase in  $A_S$ , 8.1% increase in  $A_U$ , and 11.6% increase in H-score values compared to I3D-emb-c when averaged across all three benchmark datasets. These results highlight the adaptability of JS-Siamese-c in the context of GZSL for IMU-based HAR and its robustness to the domain shift and hubness problems.

Additionally, JS-Siamese-DCE marks a significant performance improvement over JS-Siamese-c as evidenced by the substantial increases in  $A_S$ ,  $A_U$  and H-score values across all the benchmark datasets. Notably, JS-Siamese-DCE exhibits a 5.7% increase in  $A_S$ , 10.1% increase in  $A_U$ , and 9.0% increase in H-score values compared to JS-Siamese-c when averaged across all three benchmark datasets. This demonstrates the effectiveness of the DCE scheme for GZSL in the context of IMU-based HAR.

#### 5.4 Ablation study

**Impact of domain shift (skeleton joint sequences embeddings vs RGB video embeddings):** We studied the impact of domain shift on the accuracy of



the JS-Siamese architecture by utilizing RGB video embeddings extracted from the F4D model [1] instead of skeleton joint sequences extracted from the pre-trained ST-GCN model. In this experiment, we made some adjustments to the JS-Siamese architecture [1] as follows. In the case of the PAMAP2 and DaLiAc benchmark datasets, after preprocessing of the IMU data, we did not transform the IMU data to a skeleton-like representation. Instead, before the feature extraction stage, we removed 10 seconds from the start and end of the video of each activity recording to ensure that no noise is included. Following [16], we segmented the raw video data using sliding windows with a window size of 5.12 seconds and an overlap of 1 second. Within each window, we computed the mean and standard deviation values and used them as features. For UTD-MHAD, we did not use the sliding window approach since the actions in UTD-MHAD are very short; instead, we treated the entire recording as a single window. We term the above described scheme as the V-Siamese-DCE architecture.

Figure 3 depicts the impact of domain shift on the JS-Siamese architecture. On the benchmark datasets PAMAP2, DaLiAc and UTD-MHAD, V-Siamese-DCE achieved H-score values of 49.2%, 64.1% and 39.8% respectively which represent a reduction of 16%, 13.2% and 9.4% respectively when compared to corresponding H-score values attained by JS-Siamese-DCE on the same datasets. The average reduction in H-score values across all three benchmark datasets is 12.9%, highlighting the importance of resolving the domain shift problem and selecting the appropriate data modality when developing a GZSL framework.

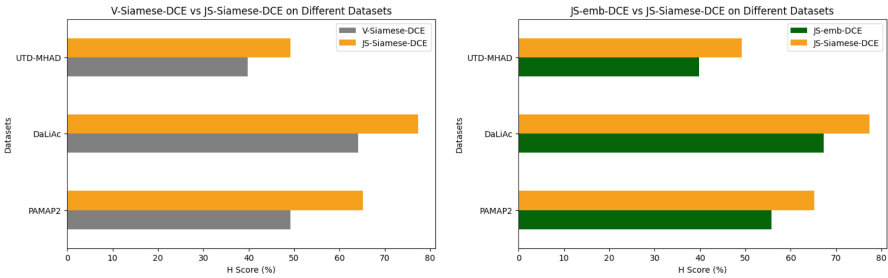


Fig. 3. Impact of domain shift (right) and impact of hubness (left)

**Impact of hubness (latent embedding space vs skeleton joint sequences embedding space):** We studied the impact of hubness on the accuracy of the JS-Siamese architecture by projecting the transformed skeleton-like representation derived from IMU data to the skeleton joint sequences space constructed using the pre-trained ST-GCN model. In this experiment, we replaced the Siamese network with a 3-layer LSTM network to process the skeleton-like representations derived from IMU data and produce the embeddings to be projected onto the skeleton joint sequences embedding space. Furthermore, we used a 3-layer LSTM network to process the joint sequences extracted from the ST-GCN model and generate the embeddings. The projection of the skeleton-like

representations from IMU data onto the skeleton joint sequences space was followed by classification in that space using the DCE procedure. We denote the resulting architecture as JS-emb-DCE. Figure 3 depicts the results of performance comparison between JS-emb-DCE and JS-Siamese-DCE. The JS-emb-DCE is observed to achieve an H-score of 55.7, 67.3 and 39.8 representing a reduction of 9.5%, 10% and 9.4% for the PAMAP2, DaLiAc and UTD-MHAD datasets respectively when compared to JS-Siamese-DCE. The average decrease in H-score across all datasets is 9.63% signifying the importance of addressing the hubness problem when designing a GZSL framework (Table 2).

**Table 2.** Impact of bias

Approach	Space	PAMAP2			DaLiAc			UTD-MHAD		
		$A_S(\%)$	$A_U(\%)$	$H$	$A_S(\%)$	$A_U(\%)$	$H$	$A_S(\%)$	$A_U(\%)$	$H$
JS-Siamese	Skeletal videos	76.4	10.1	17.8	96.5	9.8	17.8	58.7	18.1	27.6
JS-Siamese-c	Skeletal Videos	63.4	49.1	55.3	77.8	56.7	65.5	48.4	39.8	43.6
JS-Siamese-DCE	Skeletal Videos	69.8	61.2	<b>65.2</b>	85.7	67.5	<b>77.3</b>	51.4	47.3	<b>49.2</b>

**Impact of bias on JS-Siamese architecture:** We studied the impact of bias on the JS-Siamese architecture by comparing its performance with and without addressing the bias problem. Specifically, we compare the original JS-Siamese model with the JS-Siamese-DCE model, which incorporates the Dynamic Calibration Ensemble (DCE) technique to mitigate bias. Table 2 depicts the impact of bias on the JS-Siamese architecture. The average reduction in bias can be quantified by comparing the  $A_U$  and H-score values across the three datasets. For the JS-Siamese model, the average  $A_U$  across the PAMAP2, DaLiAc, and UTD-MHAD datasets is 12.7%, whereas for the JS-Siamese-DCE model, it is 58.7%. This represents an average increase of 46% in the  $A_U$  value. Similarly, the average H-score for the JS-Siamese model is 21.1%, whereas for the JS-Siamese-DCE model, it is 63.9%. This represents an average increase in H-score of 42.8%. Addressing the bias problem does come with a trade-off in the accuracy of seen classes ( $A_S$ ). For the JS-Siamese model, the average  $A_S$  value across the PAMAP2, DaLiAc, and UTD-MHAD datasets is 77.2%, whereas for the JS-Siamese-DCE model, it is 69%. This represents an average decrease in  $A_S$  of 8.2%. We believe that this reduction is an acceptable trade-off given the significant improvements in the accuracy of unseen classes ( $A_U$ ) and the overall H-score.

**Impact of loss function (contrastive loss vs. triplet loss):** We trained the Siamese network with two different loss functions, i.e., the contrastive loss function and the triplet function and noted the classification accuracy results in each case. As shown in Figure 4, the network trained with the contrastive loss function achieved a significantly higher accuracy of 92%, compared to the 79% accuracy obtained with the triplet loss function. In each case, the network employed 3 LSTM layers and was trained for 50 epochs using an Adam optimizer.

The higher accuracy achieved with the contrastive loss function indicates that it is better at creating a distinct separation between the different classes in the dataset, resulting in more accurate HAR. This is due to the inherent ability of the contrastive loss function to effectively differentiate between similar and dissimilar instances, which aligns well with the characteristics of the data used in the JS-Siamese architecture.

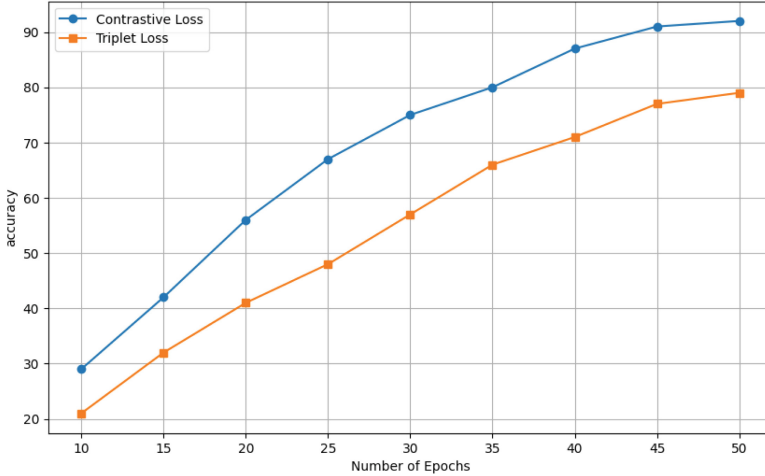
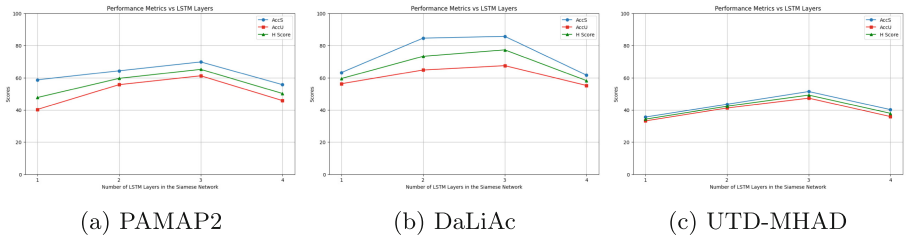


Fig. 4. Contrastive loss vs triplet loss

**Impact of the number of LSTM layers:** We systematically explored how varying the number of LSTM layers in the Siamese network influences the performance of the JS-Siamese-DCE architecture. We conducted experiments employing network configurations with 1, 2, 3 and 4 LSTM layers to assess their impact on model performance. Performance metrics, including classification accuracy for both *seen* and *unseen* classes (i.e.,  $A_S$  and  $A_U$ ), as well as the H-score, were recorded for the three benchmark datasets. Figures 5a, 5b, and 5c show the impact of varying the number of LSTM layers in the Siamese network on model performance. The goal was to determine the optimal LSTM layer configuration for maximizing the performance of the JS-Siamese-DCE architecture. For all of the three benchmark datasets, a Siamese architecture with 3 LSTM layers was observed to yield the best performance metrics.

## 6 Conclusions

In summary, this paper proposed a novel architecture for GZSL in the context of HAR using IMU data. The proposed JS-Siamese architecture tackles the critical and inherent challenges associated with traditional GZSL settings, namely,



**Fig. 5.** Impact of different number of LSTM layers on model performance

the domain shift problem, hubness problem, and bias towards *seen* classes over *unseen* classes. The advantages of the proposed JS-Siamese architecture are demonstrated through extensive experiments on various benchmark datasets, showing significant performance improvement over existing methods. Our future work will explore the incorporation of point cloud input data, acquired using a depth sensor, in the proposed architecture along with IMU and RGB video input data.

## References

1. Al-Saad, M., Ramaswamy, L., Bhandarkar, S.: F4D: Factorized 4D Convolutional Neural Network for Efficient Video-level Representation Learning. In: Proc. ICAART (2024)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proc. IEEE Conf. CVPR. pp. 7291–7299 (2017)
3. Chao, W.L., Changpinyo, S., Gong, B., Sha, F.: An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In: Proc. ECCV. pp. 52–68. Springer (2016)
4. Chen, C., Jafari, R., Kehtarnavaz, N.: UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: Proc. IEEE ICIP. pp. 168–172. IEEE (2015). <https://doi.org/10.1109/ICIP.2015.7350781>
5. Cheng, H.T., Griss, M., Davis, P., Li, J., You, D.: Towards zero-shot learning for human activity recognition using semantic attribute sequence model. In: Proc. ACM UBIComp. pp. 355–358 (2013). <https://doi.org/10.1145/2493432.2493511>
6. Cheng, H.T., Sun, F.T., Griss, M., Davis, P., Li, J., You, D.: Nuactiv: Recognizing unseen new activities using semantic attribute-based learning. In: Proc. MobiSys. pp. 361–374 (2013). <https://doi.org/10.1145/2462456.2464438>
7. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint [arXiv:1705.06950](https://arxiv.org/abs/1705.06950) (2017)
8. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Proc. NIPS **33**, 18661–18673 (2020)
9. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proc. IEEE Conf. CVPR. pp. 951–958. IEEE (2009)

10. Leutheuser, H., Schuldhaus, D., Eskofier, B.M.: Hierarchical, multi-sensor based classification of daily life activities: comparison with state-of-the-art algorithms using a benchmark dataset. *PLoS ONE* **8**(10), e75196 (2013). <https://doi.org/10.1371/journal.pone.0075196>
11. Madgwick, S., et al.: An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Report x-io and University of Bristol (UK) **25**, 113–118 (2010)
12. Matsuki, M., Lago, P., Inoue, S.: Characterizing word embeddings for zero-shot sensor-based human activity recognition. *Sensors* **19**(22), 5043 (2019)
13. Ohashi, H., Al-Naser, M., Ahmed, S., Nakamura, K., Sato, T., Dengel, A.: Attributes' importance for zero-shot pose-classification based on wearable sensors. *Sensors* **18**(8), 2485 (2018)
14. Olinski, M., Gronowicz, A., Ceccarelli, M., Cafolla, D.: Human motion characterization using wireless inertial sensors. In: *Proc. MTM Robotics*. pp. 401–408. Springer (2017). [https://doi.org/10.1007/978-3-319-45450-4\\_40](https://doi.org/10.1007/978-3-319-45450-4_40)
15. Reiss, A., Stricker, D.: Creating and benchmarking a new dataset for physical activity monitoring. In: *Proc. Intl. Conf. Pervasive Tech. Related to Asst. Environ.* pp. 1–8. New York, NY, USA (2012). <https://doi.org/10.1145/2413097.2413148>
16. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: *Intl. Symp. Wearable Comp.* pp. 108–109. IEEE (2012). <https://doi.org/10.1109/ISWC.2012.13>
17. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint [arXiv:1212.0402](https://arxiv.org/abs/1212.0402) (2012)
18. Tong, C., Ge, J., Lane, N.D.: Zero-shot learning for IMU-based activity recognition using video embeddings. *Proc. ACM IMWUT*, volume=5, number=4, pages=1–23, year=2021, publisher=ACM New York, NY, USA, <https://doi.org/10.1145/3494995>
19. Wang, W., Li, Q.: Generalized Zero-Shot Activity Recognition with Embedding-Based Method. *ACM Trans. Sensor Networks* **19**(3), 1–25 (2023). <https://doi.org/10.1145/3582690>
20. Wang, W., Miao, C., Hao, S.: Zero-shot human activity recognition via nonlinear compatibility based method. In: *Proc. Intl. Conf. Web Intell.* pp. 322–330. New York, NY, USA (2017). <https://doi.org/10.1145/3106426.3106526>
21. Wu, T., Chen, Y., Gu, Y., Wang, J., Zhang, S., Zhechen, Z.: Multi-layer cross loss model for zero-shot human activity recognition. In: *Proc. PAKDD Part I* 24. pp. 210–221. Springer International Publishing, Cham (2020)
22. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: *Proc. AAAI Conf. AI*. vol. 32 (2018)



# LightHART: Lightweight Human Activity Recognition Transformer

Syed Tousiful Haque<sup>1</sup>(✉), Jianyuan Ni<sup>1</sup>, Jingcheng Li<sup>2</sup>, Yan Yan<sup>3</sup>,  
and Anne Hee Hiong Ngu<sup>1</sup>

<sup>1</sup> Texas State University, San Marcos, TX, USA  
{bgu9,j\_n317,angu}@txstate.edu

<sup>2</sup> University of New South Wales, Sydney 2052, NSW, Australia  
jingcheng.li@unsw.edu.au

<sup>3</sup> Illinois Institute of Technology Chicago, Chicago, IL, USA  
yyan34@iit.edu

**Abstract.** Human Activity Recognition (HAR) using wearable sensors has gained significant attention due to its portability and unobtrusiveness. However, the data obtained from wearable sensors are limited to inertial data from predefined locations on the human body. In contrast, skeletal data from motion capture devices, such as the Kinect camera, offer richer information by capturing the whole body dynamics of a human action. Unfortunately, the use of skeletal data is impractical in wearable sensor-based HAR for real-world deployment. Currently, transformer neural networks, known for their self-attention mechanism, have shown effective handling of data from diverse modalities in wearable sensor-based HAR. However, the deployment of multimodal transformer on wearable devices is challenging due to their inherent large model size. We propose a Lightweight HAR Transformer (LightHART) framework that trains a unimodal Inertial Transformer (IT) network by transferring knowledge from a large multimodal transformer using a knowledge distillation approach. We evaluate the proposed framework on three public multimodal human activity datasets and compare the performance of the LightHART student model with various state-of-the-art approaches. Experimental results demonstrate that our LightHART model achieves competitive performance in terms of effectiveness and scalability with a model size of only 1.43 Mb. We are the first to deploy and validate the LightHART fall detection model on a SmartFall App running on a WearOS-compatible smartwatch showcasing its potential in advancing wearable sensor-based HAR research.

**Keywords:** Human Activity Recognition · Transformer · Knowledge Distillation · Multi-modal Learning · Wearable Devices

---

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/978-3-031-78354-8-27>.

## 1 Introduction

A wearable sensing system that can facilitate Human Activity Recognition (HAR) utilizing information extracted from diverse visual and inertial (accelerometer, gyroscope, etc.) modalities can have a significant societal impact. For example, HAR can improve elder care in assisted living centers from timely detection of falls and timely administration of medication. In addition, HAR can also revolutionize diverse context-aware applications like fitness tracking, health monitoring, and gesture recognition, just to name a few [16].

Human perceives the world in a multimodal view, automatically integrating information from multiple sensors like vision, sound, touch, etc. It is known that multimodal deep learning approaches can leverage information from multiple sources like accelerometers, gyroscopes, and visual inputs and alleviate the limitation regarding unimodal approaches via complementary information, reducing the ambiguity of activity recognition, and being robust against noisy data. While the multimodal learning model offers various benefits for HAR problem, implementing them in wearable devices is challenging due to hardware limitations in executing models of large size and the inability to acquire the visual modality continuously with on-body sensors without compromising users' privacy.

Knowledge Distillation (KD) is a potential solution that can leverage multimodal algorithms for wearable devices. KD was first introduced in [9] to distill knowledge from large models *i.e. teacher* into smaller models *i.e. student*. Initially, a large complex model is trained with data suitable for the task. These models typically had a large number of parameters and thus can achieve high accuracy by learning rich representations. Next, a smaller model is trained on the same dataset, but instead of using only the ground truth labels, it is trained to mimic the behavior of the teacher model. To improve the performance of deep learning models on HAR tasks involving vision modality, particularly when dealing with occlusion, the authors of [13] introduced a multimodal knowledge distillation approach that integrates diverse sensor information. A cross-modal knowledge distillation method is introduced in [23] that transfers knowledge from multimodal to unimodal networks. Though this work aimed to produce a model for wearable devices, the ResNet18 student network used in this research resulted in a complex model that is not usable in wearable devices. A small Distilled Mid-fusion Transformer student model is produced by [14], but the student model only works in the presence of multimodal data, which makes it inappropriate for use in portable wearable devices since it is not possible to acquire the visual data in real-time while being mobile and free of the burden to carry a specialized on-body visual sensor. Meanwhile, previous studies applied several fusion methods in building effective multimodal model [14, 22]. For instance, the work in [22] uses a late fusion, and the authors in [14] introduce a Temporal Mid Fusion. However, these fusions don't take the spatial and temporal features into account at the same time and thus can't produce an effective knowledge representation when transferring to student models.

To leverage multi-modal learning on wearable devices, we propose a Lightweight HAR Transformer (LightHART) framework that produces an Inertial

Transformer (IT), the student model, that can learn to mimic a Spatio-Temporal ConvTransformer (STConvT) teacher model. First, we train the STConvT model with data from multiple modalities (*i.e. skeleton, inertial*) and fuse the spatial and temporal information using Attention Feature Fusion. We then train the student model on only inertial data (unimodal) guided by the feature representation acquired in STConvT using knowledge distillation. This LightHART framework tries to minimize the distillation loss during its training. After training, LightHART’s student model can achieve competitive performance on three multimodal HAR datasets with a model size of only 1.43 Mb. We further tested and deployed the LightHART fall detection model (a specific type of human activity) on a SmartFall App [21] running on a WearOS-compatible smartwatch. The contributions of this paper are summarized as follows:

- We propose LightHART that generates a lightweight transformer model running on inertial modalities only. To our knowledge, this is the first study conducting a knowledge distillation process from a skeleton-to-inertial domain using an unimodal Transformer model which is lightweight.
- We propose a STConvT model with Attention Feature Fusion that can produce better feature representation aligning both spatial and temporal information.
- We demonstrated the effectiveness and generalization ability of the proposed LightHART method on three public datasets.
- We are the first to test and deploy the LightHART fall detection model on a real-world fall detection App to demonstrate its potential in advancing wearable sensor-based HAR research.

Our paper is organized as follows. In the related work in Section 2, we describe some background work on human activity recognition and the motivation behind choosing a transformer-based architecture. Next, we present the methodology and the architecture of LightHART in Section 3. We outline the setup of the Spatial and Temporal encoder blocks and the Attention Feature Fusion strategy. In Section 4, we describe the dataset used, the experimental setup, and the evaluation protocol used. In Section 5, we compare the performance of LightHART with other SOTA approaches. In Section 6, we conduct ablation studies to showcase the effectiveness of our fusion strategy and the spatial block. Finally, we discuss the implications of our findings and future directions for our work in the conclusion section.

## 2 Related Work

**Human Activity Recognition:** HAR is used to detect and classify human activities under appropriate labels. An activity refers to the collective movement of parts of the body to complete a task. For example, moving the head in negation is a gesture, and walking, jumping, and hand waving are activities [27]. The approaches to resolving the human activity recognition task can be divided into three types: vision-based HAR [2], sensor-based HAR [8], and multimodal HAR



[14]. A wide spectrum of methods, ranging from traditional machine learning, rule-based, to deep learning methods have been used for HAR over the years.

An extensive comparison among K-Nearest Neighbor (KNN), Support Vector Machines (SVM), Gaussian Mixture Models (GMM), and Hidden Markov Models (HMM) for wearable sensor-based HAR is discussed in [1]. The early traditional machine learning approaches depend on features built by domain experts and can't efficiently differentiate between very similar activities such as walking upstairs and walking downstairs [26]. RNN, LSTM, and CNN are unimodal deep learning networks that have become popular in recent years and have achieved state-of-the-art in recognizing different HAR tasks. For example, an ensemble Recurrent Neural Network (RNN) method has been used in [17] to do fall detection from wearable devices. Multiple other research works such as those in [19, 25, 26] have used LSTM and a hybrid CNN-LSTM network for HAR.

Wearable devices using unimodal data have shown the promise of bringing personalized health monitoring closer to consumers [20]. For example, smartwatches like the Apple Series, which feature built-in "hard fall" detection and ECG monitoring apps, are a viable platform for digital health applications when paired with a smartphone. However, unimodal deep learning methods using data from wearable devices have certain limitations [12, 30]. Data produced by wearable sensors can be noisy, lack contextual information, and face difficulties discriminating among activities producing similar patterns. For example, if a person is wearing a watch on the left wrist and the left wrist does not move during a fall, the fall will be missed. Video or skeleton modalities can provide complementary and contextual information to unimodal data from wearable devices for better recognition of human activities. To capture information from both spatial and temporal domains, the authors in [16] introduced a multimodal network called AttnSense. DanHar framework was proposed in [7] to blend channel attention and temporal attention with a CNN model. However, none of the above multimodal models have a model size that is small enough for real-world deployment to a wearable device.

**Transformer:** Deep learning methods like LSTM and CNN have some inherent problems when used for HAR. Although LSTM can handle temporal dynamics in long sequences of data from human activities, their singular perception limits them in capturing complex patterns that require multiple viewpoints. Convolutional Neural Networks (CNNs) are primarily designed to extract local spatial patterns within data. By leveraging multiple layers, they can also capture more complex and global spatial features. However, CNNs are inherently limited in their ability to process temporal information. The continuous HAR signal patterns are more distinguishable when seen from a global temporal viewpoint. Transformer [28] possesses a global viewpoint courtesy of its self-attention layer, and the multiple heads in self-attention help to create multiple viewpoints. Transformer has already been used successfully in NLP, Computer Vision, Recommendation Systems, and many others. It also has been used in HAR. For

example, the authors in [31] used a two-stream Transformer network to capture both spatial and temporal features from inertial data.

However, these multimodal networks aren't suitable for deployment in wearable sensors due to the unavailability of visual modalities in real-time [13, 15, 23]. Moreover, the constraints of computation power of wearable devices preclude the deployment of the usually large multi-modal learning model.

To the best of our knowledge, only a few studies by [6, 10, 32] have conducted efficient experiments on lightweight transformer-based architecture in HAR domain.

### 3 Methodology

In this paper, we introduce our LightHART framework that produces a lightweight (Inertial Transformer) student model from the knowledge distillation process that only uses inertial data and still maintains similar accuracy as the multimodal teacher model. An STConvT network works as a teacher by extracting the salient spatial and temporal features and using an Attention Feature Fusion to combine features from skeleton and inertia modalities effectively.

Figure 1 gives the overview of the knowledge distillation process that distillates knowledge from a multimodal teacher model to an unimodal student model. First, we train a multimodal STConvT teacher network with skeleton and inertial data. The input from different modalities is segmented using the sliding window technique described in [34]. We then add a learnable positional embedding to each of the modalities to preserve the positional information. A Spatial Block consisting of two convolutional layers extracts accurate spatial information from the skeleton data.

The output of the Spatial Block is divided into patches and passed on to a Temporal block which leverages ViT architecture [5]. The Temporal Block consists of two Transformer Encoders that apply a multi-head self-attention mechanism [28] on the patches to extract the salient temporal features while preserving spatial information. On the other hand, inertial data is passed to a separate Temporal Block. The features from the intermediate Transformer Encoder dedicated to the skeleton and inertial data are fused using Attention Feature Fusion and passed to an MLP layer for final prediction. Finally, a knowledge distillation procedure is used to transfer the feature representation learned by the teacher module to the student's Inertial Transformer(IT) that works on inertial data only. Our IT also adopts the ViT architecture [5]. In the following, we elaborate on the framework to produce the lightweight student model using the knowledge distillation procedure.

#### 3.1 Inertial Transformer (IT)

The original transformer model consists of an encoder and a decoder. The encoder generates embeddings from the input, while the decoder uses these embeddings to produce output in a different language. However, for activity

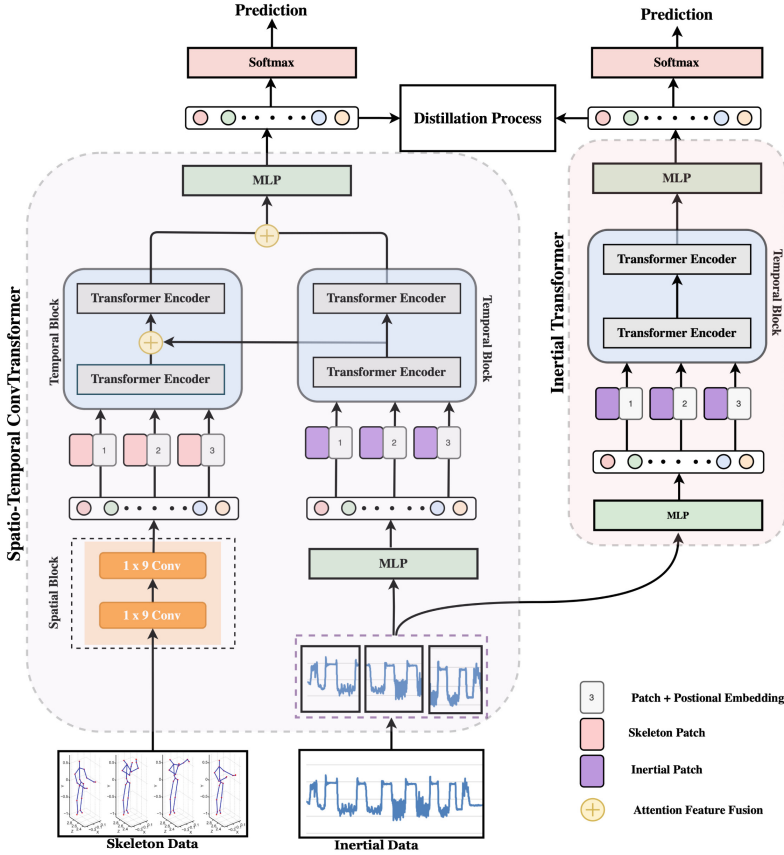


Fig. 1. LightHART framework with STConvT as teacher and IT as student.

recognition, only the encoder is needed to extract both spatial and temporal information.

In ViT, an image’s input is first segmented into patches. We can think of the inertial data as 2-D images with shape  $(W, C_{iner})$  where  $W$  is the window size and  $C_{iner}$  is the number of channels of inertial data. The input for IT  $x \in \mathbb{R}^{(N \times (W \times C_{iner}) / P)}$  is reshaped into a sequence of patches, where  $N$  is the number of patches and  $P$  is the patch size. The IT uses a constant embedding size of  $D$  through all its layers. The patches are transformed to  $D$  dimension using a linear layer (Eq. 1). A learnable class token is appended at the start of the sequence of embedding patches ( $z_0^0 = x_{class}$ ) whose state at the output of the Transformer’s encoder  $z_L^0$  serves as the inertial data representation  $y$ . A learnable one-dimensional positional embedding  $E_{pos}$  is added to the patch embeddings. The Transformer encoder [28] comprising of interleaved layers of multiheaded self-attention (MSA) and MLP blocks is applied to the patches. Layer normalization (LN) is employed preceding each block for stabilized train-

ing, with residual connections following each block. The residual connection was used to avoid a vanishing gradient and ensure a direct flow of information. The class token of the last encoder block output is then passed to the MLP head with the softmax activation to get the final prediction.

To keep the network small, we construct it with only two Transformer encoder blocks with small embedding dimensions in the student’s IT.

$$z_0 = [x_{class}; x_p^1 E; x_p^2; \dots; x_p^N E;] + E_{pos} \quad E \in \mathbb{R}^{((W \times C_{iner})/P) \times D} \quad (1)$$

### 3.2 Spatio-Temporal ConvTransformer

The Spatio-Temporal ConvTransformer is made up of three important parts: 1. *Spatial Block*, 2. *Temporal Blocks*, and 3. *Attention Feature Fusion* that helps it to analyze both spatial and temporal information effectively.

**Spatial Block:** The Spatial Block is depicted in Fig 1 in orange color. This module is in charge of dealing with the spatial details found in skeleton data. It uses two 2-dimensional (2D) convolution layers that could effectively extract the relationships between nearby joints. These layers have a special property called translation invariance inductive bias, making them particularly effective at processing spatial information. Let  $x_{SK} \in \mathbb{R}^{(C_{SK}, J_{SK}, W_{SK})}$  is the skeleton input to the Spatial Block where  $C_{SK}$  is the channels of skeleton data,  $J_{SK}$  is the number of predefined joints and  $W_{SK}$  is the size of the window. The 2D Convolution layers in the Spatial Block take in an input of  $(C_{in}, H, W)$  where  $C_{in}$  is the number of channels,  $H$  is the height of input and  $W$  is width. To process the skeleton data with 2D Convolution Layer we set  $C_{in} = C_{SK}$ ,  $H = J_{SK}$ , and  $W = W_{SK}$ . Both the convolution layers had a filter shape of  $(1, 9)$  to gather spatial information from three adjacent joints. The Spatial Block (SP) produces an output  $s_p$  of shape  $(C_{out}, H_{out}, W)$ , where  $C_{out}$  is the output channel size and  $H_{out}$  is the output height as of Eq. 2. The output of the Spatial Block is then reshaped to  $(N, C_{out} \times H_{out} \times (W/P))$  where  $N$  is the number of patches and  $P$  is the size of the patches.

$$s_p = \mathbf{SP}(X_{skl}) \quad X_{skl} \in \mathbb{R}^{(C_{SK}, J_{SK}, W_{SK})} \quad (2)$$

**Temporal Block:** The Temporal Block has a structure that is the same as the IT. So, the sequence of skeleton patches  $s_p$  is transformed into  $z_0^{skl} \in \mathbb{R}^{(B, P, D)}$  (Eq. 4) where  $D$  remains constant all across the network. Creating patches from the embedding will help the Temporal Block to process temporal information together [5]. We also process the inertial data with a Temporal Block. Let  $X_{iner} \in \mathbb{R}^{(W \times C_{iner})}$  be the inertial data. This inertial is then reshaped to  $i_p \in \mathbb{R}^{(N \times (W \times C_{iner})/P)}$ . To match the dimension of the Transformer Encoder the input is transformed to  $x \in \mathbb{R}^{(N \times D)}$  and 1-D learnable positional embedding  $E_{pos}$  and class embedding  $i_{class}$  was added (Eq. 3). The processed inertial data and output from the Spatial Block then go through the first Encoder on two

different Temporal Blocks as shown in Fig 1 and produce embedding  $z_1^{skl}$  and  $z_1^{iner}$  (Eq. 5).

$$z_0^{iner} = [i_{class}; i_p^1 E; i_p^2; \dots; i_p^N E;] + E_{pos} \quad E \in \mathbb{R}^{((W \times C_{iner})/P) \times D} \quad (3)$$

$$z_0^{skl} = [s_{class}; s_p^1 E; s_p^2; \dots; s_p^N E;] + E_{pos} \quad E \in \mathbb{R}^{((C_{out} \times H_{out} \times W)/P) \times D} \quad (4)$$

$$z_1^m = \mathbf{Encoder}(z_0^m) \quad m \in (iner, skl) \quad (5)$$

**Attention Feature Fusion**  $z_1^{skl}$  and  $z_1^{iner}$  are then added together to produce  $z_1^{comb}$  (Eq. 6). This fusion purpose is named as **Attention Feature Fusion(AFF)** as the output of transformer encoder layers dedicated to different modalities are fused. AFF merges complementary information from temporally aligned patches of different modalities. This fusion in terms helps the subsequent self-attention layer(MSA) in better exploring the relation between patches. For all subsequent layers,  $z_l^{comb}$  is produced by fusing  $z_{l-1}^{comb}$  and  $z_{l-1}^{iner}$  (Eq. 7). The final prediction  $y$  is generated by passing the class token  $z_L^{comb_0}$  of the L-th encoder block (last) through an MLP layer (Eq. 8). A softmax function is used on the output of the MLP layer to produce the class predictions.

$$z_1^{comb} = \mathbf{Encoder}(z_1^{skl} + z_1^{iner}) \quad (6)$$

$$z_l^{comb} = \mathbf{Encoder}(z_{l-1}^{comb} + z_{l-1}^{iner}) \quad (7)$$

$$y = \text{softmax}(\mathbf{MLP}(z_L^{comb_0})) \quad (8)$$

### 3.3 Multimodal to Unimodal Knowledge Distillation

The knowledge distillation begins after we finish training the STConvT with skeleton and inertial data. During knowledge distillation, a teacher's STConvT takes multimodal (skeleton & inertial) data as input and the student's IT takes only the inertial data. In general, neural networks produce a class probability by taking the logits and passing it through a softmax function  $p_i = \text{softmax}(z^i)$ . But, the knowledge distillation method in [9] used a soft prediction with parameter Temperature (T) (Eq. 9). The higher the temperature the softer the prediction. Both the teacher and the student produce soft predictions  $P_{teacher}$  and  $P_{student}$ . These soft predictions are then compared using a KL-Divergence Loss (Eq. 10). The entropy between the ground truth  $y^{gt}$  and the student's (IT) final prediction  $y^{stud}$  is measured using a cross-entropy loss and added with the KL-Divergence loss to get the knowledge distillation loss  $\mathcal{L}_{KD}$  (Eq. 11). The student model tries to mimic the teacher's prediction by minimizing this loss during its training.

$$p_i = \frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}} \quad (9)$$

$$\mathcal{L}_{kl}(P_{student}, P_{teacher}) = \sum_i P_{student,i} \log \frac{P_{student,i}}{P_{teacher,i}} \quad (10)$$

$$\mathcal{L}_{KD} = \mathcal{L}_{cross}(y^{gt}, y^{stud}) + \mathcal{L}_{kl}(P_{teacher}, P_{student}) \quad (11)$$

## 4 Experiments

### 4.1 Datasets

We evaluated the LightHART’s performance on three human activity datasets. UTD-MHAD and Berkeley-MHAD are a few of the mainstream multimodal human activity recognition datasets publicly available. SmartFallMM is another multimodal human activity recognition dataset developed in our lab with a specific focus on fall detection.

The UTD-MHAD dataset [3] was collected using a single Kinect camera and one wearable inertial sensor. The Kinect camera captures full-body visual data during activities, while the inertial sensor records acceleration, gyroscope, and magnetometer data. The sensor was placed on the subject’s right wrist or thigh, depending on whether the action primarily involved the arm or leg. The use of only a Kinect camera and inertial sensor is due to their low cost and non-intrusive nature. The dataset includes 27 actions performed by 8 subjects (4 males and 4 females), with each action repeated 4 times, resulting in 861 samples after excluding corrupted ones.

The Berkeley-MHAD dataset [24] consists of temporally synchronized and geometrically calibrated data from an optical mocap system, multi-baseline stereo cameras from multiple views, depth sensors, accelerometers, and microphones. We used the accelerometer data collected from the left wrist for our experiment. It contains 11 actions performed by 7 male and 5 female subjects in the range of 23-30 years of age except for one elderly subject. All the subjects performed 5 repetitions of each action, yielding about 660 samples which correspond to about 82 minutes of total recording time.

The SmartFallMM<sup>1</sup> multi-modal dataset comprises data from two distinct modalities, collected using four different types of devices. The skeleton data was gathered using three Azure Kinect cameras. Additionally, accelerometer and gyroscope data were obtained from three types of inertial sensors: Meta sensors (from MBIENT), a Huawei Smartwatch running WearOS, and a Google Nexus phone. This dataset includes a total of 14 activities, performed by 36 participants. Among these activities, 9 are Activities of Daily Life (ADL), and 5 are different fall activities, resulting in a total of 1,134 activity trials, and only 11 participants could perform fall activities. We used the accelerometer data sensed from Huawei SmartWatch and the skeleton data for our experiments.

<sup>1</sup> Url: <https://anonymous.4open.science/r/smartfallmm-4588>

## 4.2 Evaluation Protocol

For the UTD-MHAD dataset, we follow the established evaluation protocol outlined in the original paper [3]. Specifically, subjects with odd-numbered identifiers (1, 3, 5, 7) are designated for training purposes, while subjects with even-numbered identifiers (2, 4, 6, 8) are reserved for testing. Given the limited size of the dataset, this approach serves to maintain a balance between the sizes of the training and testing datasets. Moreover, the segmentation based on person IDs serves the dual purpose of preventing data leakage and ensuring the integrity of the evaluation process. We adhere to the evaluation protocol outlined in the original paper [24] for Berkeley-MHAD. The training dataset comprises of first 7 persons’ data while the testing dataset consists of the last 5 persons’ data.

We performed recognition of fall-related activities on SmartFallMM dataset with real-world testing and evaluation in mind, as we already have a fall detection system developed for a wearable device [21]. We used the first 9 persons’ data for training and the last 2 persons’ data for testing. After training an offline student IT model with LightHART, we deploy this model to a Huawei Smartwatch running the SmartFall App for real-time evaluation. Two student participants are recruited under IRB 9461 for the real-time evaluation. They performed all 9 ADLs and 5 Fall activities five times each activity wearing the smartwatch.

## 4.3 Experimental Setup

The inertial modality may contain multiple streams (e.g. the accelerometer and gyroscope) of data. Despite the presence of different streams, we consider them as a single modality since they are all time-series data. Skeleton data is sensed as a sequence of time-series (accelerometer) data from multiple skeletal nodes. Both skeleton and inertial data have variable lengths across activity trials and different sampling rates. To optimize training, we equalized the sampling rates and extracted synchronized windows of size 64 from both skeleton and inertial modalities, with a 10-timestamp overlap between windows. The STConvT architecture consists of 2 consecutive Convolution layers with both having a filter size of 9 to facilitate the extraction of spatial information from adjacent joints. The two Temporal Blocks had two Transformer encoders each with an input dimension of 32. To optimize the model, we employed the SGD optimizer with a learning rate set at 0.0025 and utilized the knowledge distillation loss (Eq. 11) function during the training phase.

# 5 Studies and Results

## 5.1 Evaluations and Comparisons

We compared our LightHART’s performance with other state-of-the-art multimodal transformers with knowledge distillation-based methods using inertial and skeleton data as input. Table 1 and 2 show the experimental results on UTD-MHAD and Berkeley-MHAD respectively. We evaluated SmartFallMM

mainly for fall detection activities and is not included in this table. The inertial data from UTD-MHAD had two streams (accelerometer and gyroscope). We compared the performance of LightHART with multimodal transformer models like CrossVit [33], DMFT [14] and TokenFusion [29]. LightHART outperformed these transformer-based methods as it consecutively gains 8.67% and 14.44%, over TokenFusion [29], CrossVit [33]. Though DMFT [14] has a higher accuracy of 92.12%, it’s worth mentioning that it had a complex architecture with 262.2× larger model size than the student model trained with LightHART which makes it infeasible for deployment in wearable devices. The increased accuracy of LightHART is primarily due to the knowledge distillation method. Before knowledge distillation, the accuracy of LightHART student’s model was 73.618 % on UTD-MHAD dataset and the teacher Spatial-Temporal ConvTransformer had an accuracy of 89.81%.

**Table 1.** Performance comparison on the UTD-MHAD dataset. S: Skeleton, D: Depth, I: Inertial, aug: augmentation.

Method	Modality Combination	Accuracy(%)
UTD-MHAD [3]	I + D	81.86
Gimme Signals [18]	I + S	76.13
Gimme Signals [18]	I + S(aug)	86.53
TokenFusion [29]	I + S	78.89
CrossViT [33]	I + S	75.37
MobileHART(XS) [6]	I	77.52
DMFT [14]	I+S	<b>92.12</b>
LightHART(Teacher)	I + S	89.81
LightHART (Student)	I	73.62
LightHART(KD)	I	<b>87.56 (13.94 ↑)</b>

But after the knowledge distillation, the accuracy of the student model went up to 87.56% which is a 13.942% increase in accuracy. The LightHART student model also has an 10.037% accuracy gain over MobileHART(XS) [6] - a lightweight Transformer model - which further supports the effectiveness of our LightHART framework.

The gap between teacher and student is as small as 2.25% which demonstrates that the STConvT supported by Attention Feature Fusion creates feature representations that the student’s uni-modal IT (Inertial Transformer) can easily mimic.

Similar trends are observed in the case of the Berkeley-MHAD dataset. The inertial modality had only the accelerometer stream for this dataset. LightHART outperformed multimodal Transformer networks like TokenFusion [29] and



**Table 2.** Performance comparison on the Berkeley-MHAD dataset. S: Skeleton, D: Depth, I: Inertial

Method	Modality Combination	Accuracy(%)
MMhar-EnsembleNet [4]	I + D	81.86
TokenFusion [29]	I + S	79.91
CrossVit [33]	I + S	75.37
DMFT [14]	I + S	78.18
LightHART (Teacher)	I + S	<b>85.69</b>
LightHART (student)	I	80.33
LightHART(KD)	I	81.93(1.60 $\uparrow$ )

Cross-Vit [33] and DMFT [14]<sup>2</sup> as it consecutively gains 3.04% and 6.56% and 3.75% . The knowledge distillation method effectively increased the accuracy of the student model by 1.6%. The gap between teacher and student was 3.76%. The accuracy gain after knowledge distillation was 1.6% which is lower than the UTD-MHAD dataset. This was due to the absence of a gyroscope stream in inertial data as gyroscopes provide much-needed information about angular velocity. We couldn't compare the results with MobileHART(XS) [6] as it required both gyroscope and accelerometer modalities.

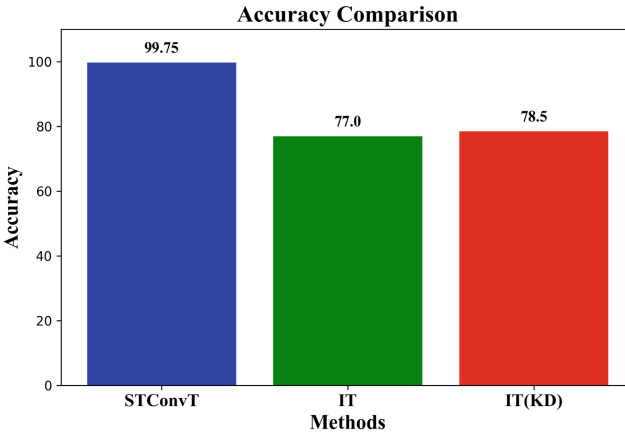
**Fig. 2.** Accuracy Comparison for Fall Detection Task of SmartfallMM dataset

Fig. 2 illustrates the performance comparison of STConvT, IT, and IT with KD on the SmartFallMM dataset. The teacher model, STConvT, achieved an

<sup>2</sup> DMFT wasn't originally evaluated on Berkley-MHAD datasets. We trained this model for 250 epochs for Berkley-MHAD to provide the same training time for fair comparison

accuracy of 99.75%, while the student IT model of LightHART had an accuracy of 77.0% before applying KD. By employing STConvT as the teacher during the knowledge distillation process, the accuracy of the IT model increased by 1.50% for fall detection.

Table. 3 shows the model size comparison of different multimodal Transformer models. The student model generated using LightHART had a model size of 1.43 Mb which is  $262.2\times$  smaller than DMFT [14] which has a model size of 375 Mb. The DMFT uses a ResNet50 pre-trained model size of 98 Mb. Even if they used an architecture without the ResNet50, the model size would still be  $193.7\times$  larger than the student model generated by LightHART. CrossVit [33] and MobileHART(XS) also have  $425\times$  and  $7.23\times$  larger model sizes compared to our student model. Only TokenFusion [29] has a smaller model size than our student network. However, this smaller model size also compromises the accuracy as it drops to 78.89% for UTD-MHAD and 79.91% for Berkeley-MHAD. Overall, only our student model can maintain competitive performance while reducing the model size.

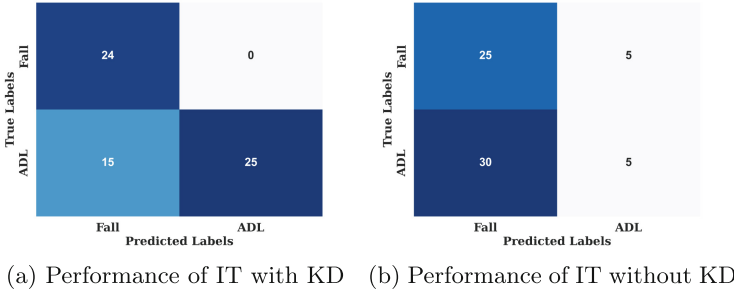
**Table 3.** Model Size comparison for different Transformer models

Modalities	Model	Model size(mb)
I	LightHART	1.43
I + S	TokenFusion [29]	<b>.68</b>
I + S	CrossVit [33]	608.09
I + R + S	DMFT [14]	375
I	MobileHART(XS) [6]	10.36

## 5.2 Performance on Wearable Devices

We ported two different IT models to run on a smartwatch, one generated by LightHART and the other purely based on uni-modal accelerometer data without knowledge distillation to observe the average inference time and performance. Both of our IT models running on the device could make an inference in .4459 ms to .8428 ms for a stream of data with a duration of 4 seconds compared to 1 to 13 ms for 2.56 seconds duration of data using MobileHART(XS) [6]. The LightHART student IT model’s performance improvement in fall detection task after training with KD can also be observed in Figure 3. Though both the models have a similar number of True Positive detection of 24 and 25, the IT model trained without KD cannot differentiate the intrinsic patterns between ADL and Fall activities as it can only detect 5 of 35 ADL activities accurately compared to 25 out of 35 of the student model trained with KD. The accuracy of the model without KD drops by 31.69% and becomes 45.31% during the on-device evaluation. The student’s model trained with KD can maintain similar

accuracy with on-device evaluation as its accuracy only becomes 76.56% which represents only a 1.94% drop. This on-device performance comparison shows models trained with KD can help maintain better performance.



**Fig. 3.** Confusion matrices for on device performance of LightHART(student) with and without Knowledge Distillation

## 6 Ablation Studies

### 6.1 Effectiveness of Attention Feature Fusion

Table 4 shows the effectiveness of Attention Feature Fusion(AFF). For this experiment, we used the SimpleFusion [11], TokenFusion by [29], CrossView Fusion by [33] and Attention Feature Fusion(AFF) with our STConvT to observe which fusion methods have the most impact on the student model’s accuracy. The result shows that the teacher model using AFF has a student model with the highest accuracy of 87.56%. Though the teacher network with CrossView Fusion had better accuracy, the representation was complex for a lightweight student model to mimic. Thus, the student had the lowest accuracy of 69.47%

**Table 4.** Performance comparison of different fusion methods on UTD-Mhad dataset

Method	Teacher Accuracy(%)	KD Accuracy(%)
SimpleFusion [11]	87.68	84.36
TokenFusion [29]	85.00	70.04
CrossView Fusion [33]	<b>90.0</b>	69.47
AFF	89.81	<b>87.56</b>

## 6.2 Effectiveness of Convolution Spatial Block

Table 5 shows the impact of the Convolution Spatial Block. First, we changed the Spatial Block to a Transformer-like architecture with 2 encoders. The accuracy dropped to 77.12% in comparison to 89.81% for the Convolution Spatial Block. This shows that the Convolutional layers with an inductive bias for spatial information outperform vanilla transformers. On the other hand, a network without Spatial Block had an accuracy of 80.25% which is 9.56% lower than a model with Convolution Spatial Block.

**Table 5.** Performance comparison with and w/o Convolution Spatial Block

Method	Teacher Accuracy(%)
Transformer SB	77.12
W/O SB	80.25
<b>Convolution SB</b>	<b>89.81</b>

A supplementary study on the effectiveness of the Temporal Block is presented in Table 1 of the supplementary materials.

## 7 Conclusion

In this paper, we propose a LightHART network architecture to generate a lightweight transformer model (student) using unimodal inertial data that has a very small model size while retaining similar accuracy as the complex multimodal transformer (teacher) network in the case of UTD and Berkeley datasets. With SmartFallMM dataset, we show that the IT model with KD performs better than the one without. The experimental results also demonstrate that our lightweight student model with a model size of 1.43 Mb can achieve competitive performance as compared to other student models distilled from state-of-the-art multimodal learning frameworks. We further tested and deployed the LightHART student’s model on a wearable smartwatch device running a fall detection App. The real-world testing of the model using two participants demonstrates the better performance of a uni-modal fall detection trained using a knowledge distillation approach. However, while we have demonstrated that a lightweight LightHART model can be deployed successfully on the device that outperforms the model without KD, there is still a considerable performance gap between the teacher and student model in LightHART which we believe can be reduced by adopting more advanced knowledge distillation methods. Furthermore, using the Smart-FallMM dataset, the fall detection model trained with KD still needs to be optimized to reduce the high False Positive ratio for practical use.

**Acknowledgment.** We thank the National Science Foundation for funding the research under the NSF-SCH (21223749).

## References

1. Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., Amirat, Y.: Physical human activity recognition using wearable sensors. *Sensors* **15**(12), 31314–31338 (2015)
2. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. *TPAMI* **24**(8), 1091–1104 (2002)
3. Chen, C., Jafari, R., Kehtarnavaz, N.: Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: *ICIP*. pp. 168–172. IEEE (2015)
4. Das, A., Sil, P., Singh, P.K., Bhateja, V., Sarkar, R.: Mmhar-ensemnet: a multimodal human activity recognition model. *IEEE Sens. J.* **21**(10), 11569–11576 (2020)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
6. EK, S., Portet, F., Lalanda, P.: Lightweight transformers for human activity recognition on mobile devices. arXiv preprint [arXiv:2209.11750](https://arxiv.org/abs/2209.11750) (2022)
7. Gao, W., Zhang, L., Teng, Q., He, J., Wu, H.: Danhar: Dual attention network for multimodal human activity recognition using wearable sensors. *Appl. Soft Comput.* **111**, 107728 (2021)
8. Han, J., Bhanu, B.: Human activity recognition in thermal infrared imagery. In: *CVPR Workshops*. pp. 17–17. IEEE (2005)
9. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)
10. Huan, S., Wang, Z., Wang, X., Wu, L., Yang, X., Huang, H., Dai, G.E.: A lightweight hybrid vision transformer network for radar-based human activity recognition. *Sci. Rep.* **13**(1), 17996 (2023)
11. Ijaz, M., Diaz, R., Chen, C.: Multimodal transformer for nursing activity recognition. In: *CVPR*. pp. 2065–2074 (2022)
12. Islam, M.M., Nooruddin, S., Karray, F., Muhammad, G.: Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things. *Information Fusion* **94**, 17–31 (2023)
13. Kong, Q., Wu, Z., Deng, Z., Klinkigt, M., Tong, B., Murakami, T.: Mmact: A large-scale dataset for cross modal human action understanding. In: *ICCV*. pp. 8658–8667 (2019)
14. Li, J., Yao, L., Li, B., Sammut, C.: Distilled mid-fusion transformer networks for multi-modal human activity recognition. arXiv preprint [arXiv:2305.03810](https://arxiv.org/abs/2305.03810) (2023)
15. Liu, Y., Wang, K., Li, G., Lin, L.: Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition. *TIP* **30**, 5573–5588 (2021)
16. Ma, H., Li, W., Zhang, X., Gao, S., Lu, S.: Attnsense: Multi-level attention mechanism for multimodal human activity recognition. In: *IJCAI*. pp. 3109–3115 (2019)
17. Mauldin, T., Ngu, A.H., Metsis, V., Canby, M.E.: Ensemble deep learning on wearables using small datasets. *ACM Trans. Comput. Healthcare* **2**(1). <https://doi.org/10.1145/3428666>, <https://doi.org/10.1145/3428666> (dec 2021).
18. Memmesheimer, R., Theisen, N., Paulus, D.: Gimme signals: Discriminative signal encoding for multimodal activity recognition. In: *IROS*. pp. 10394–10401. IEEE (2020)

19. Mutegeki, R., Han, D.S.: A cnn-lstm approach to human activity recognition. In: ICAIIC. pp. 362–366. IEEE (2020)
20. Ngu, A.H., Metsis, V., Coyne, S., Chung, B., Pai, R., Chang, J.: Personalized fall detection system. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp. 1–7. IEEE (2020)
21. Ngu, A.H., Yasmin, A., Mahmud, T., Mahmood, A., Sheng, Q.Z.: P-fall: Personalization pipeline for fall detection. In: Proceedings of the 8th ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies. pp. 173–174 (2023)
22. Ni, J., Ngu, A.H., Yan, Y.: Progressive cross-modal knowledge distillation for human action recognition. In: ACM MM. pp. 5903–5912 (2022)
23. Ni, J., Sarbajna, R., Liu, Y., Ngu, A.H., Yan, Y.: Cross-modal knowledge distillation for vision-to-sensor action recognition. In: ICASSP. pp. 4448–4452. IEEE (2022)
24. Offi, F., Chaudhry, R., Kurillo, G., Vidal, R., Bajcsy, R.: Berkeley mhad: A comprehensive multimodal human action database. In: WACV. pp. 53–60. IEEE (2013)
25. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **16**(1), 115 (2016)
26. Ronao, C.A., Cho, S.B.: Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **59**, 235–244 (2016)
27. Saleem, G., Bajwa, U.I., Raza, R.H.: Toward human activity recognition: a survey. *Neural Comput. Appl.* **35**(5), 4145–4182 (2023)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *NeurIPS* **30** (2017)
29. Wang, Y., Chen, X., Cao, L., Huang, W., Sun, F., Wang, Y.: Multimodal token fusion for vision transformers. In: CVPR. pp. 12186–12195 (2022)
30. Wu, Q., Huang, Q., Li, X.: Multimodal human action recognition based on spatio-temporal action representation recognition model. *Multimedia Tools and Applications* **82**(11), 16409–16430 (2023)
31. Xiao, S., Wang, S., Huang, Z., Wang, Y., Jiang, H.: Two-stream transformer network for sensor-based human activity recognition. *Neurocomputing* **512**, 253–268 (2022)
32. Xu, H., Zhou, P., Tan, R., Li, M., Shen, G.: Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In: Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems. pp. 220–233 (2021)
33. Yan, S., Xiong, X., Arnab, A., Lu, Z., Zhang, M., Sun, C., Schmid, C.: Multiview transformers for video recognition. In: CVPR. pp. 3333–3343 (2022)
34. Zhang, Y., Wang, L., Chen, H., Tian, A., Zhou, S., Guo, Y.: If-convtransformer: A framework for human activity recognition using imu fusion and convtransformer. *IMWUT* **6**(2), 1–26 (2022)



# Robust Leaf Detection using Shape Priors within Smaller Datasets

Debojyoti Misra<sup>(✉)</sup> and Tushar Sandhan

Indian Institute of Technology Kanpur, Kanpur, India  
{debojyotim22,sandhan}@iitk.ac.in

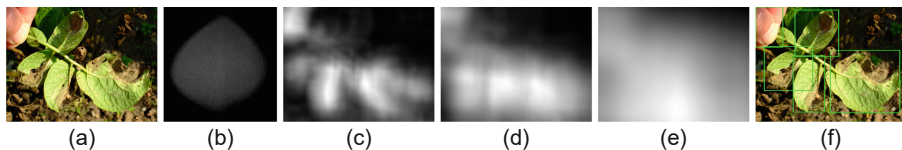
**Abstract.** Our brain can process visual information, which helps understanding the shape of an object. If never seen before, an accurate description of the shape can help ease the task for a human who is looking for an object. Data scarcity in agriculture is primarily due to the labor-intensive and cost-intensive nature of collecting, as well as the requirement for expertise to label them. Our task of leaf detection has only one kind of object, which has some general shape features. To make the task of learning easier from a comparatively smaller dataset, we automatically learn shape prototypes from leaves and use them as templates to generate shape-specific features to incorporate prior knowledge into the neural network. We use this method to generate prototypes from the Plant Village dataset and use them for detection in the Plant-Doc dataset to improve the mean average precision (mAP) by 3% over the state-of-the-art Faster-RCNN model. These kinds of experiments show the cross-dataset generalizability of the proposed method.

**Keywords:** Template matching · Feature fusion · Shape prior

## 1 Introduction

Object detection in computer vision is known for its dependency on a large amount of training data to produce high-quality performance. In domains like agriculture, it has been a challenge to gather data to train reliable models due to the labour and cost intensiveness, required expertise in labeling and the seasonal nature of agriculture. This study focuses on one such problem, namely leaf detection in field images.

Plant disease detection from leaf images has been in the research scenario for some time now. Several works [1–3] on the Plant Village [4] dataset have shown remarkable results in disease classification tasks. The Plant Village [4] dataset consists of 61486 leaf images taken in laboratories with only one leaf present in the image with proper illumination. For that reason, models trained on this dataset fail when applied to fields. Images taken in fields contain multiple leaves with other natural elements like branches, flowers, fruits, insects, or soil. To remove these kinds of noises from images, researchers have framed it as an object detection [6–8] task where leaf is the common object of interest and different diseases are of different classes. In [6], the author shows how a secondary



**Fig. 1.** In the figure we can see (a)input image and (b)a learned leaf prototype; (c), (d) and (e) are prior matching results of the input image with different scales( $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ ) of the leaf prototype; in (f) we can see the predicted bounding boxes from our model drawn on the input image

classification unit trained for plant disease classification can improve the classification part of the detection. Models proposed in [1–3] can be fine-tuned and used for efficient classification given that leaves have been accurately identified and localised in field images.

Detecting leaves accurately is a very challenging task. Leaves come in all kinds of shapes, scales, textures, and colours, resulting in high intra-class variation. Other than that, in field conditions, the leaves are very densely populated, sometimes creating occlusion in vision. The occlusion of the object increases the challenge of detection significantly. Also, some parts of the leaves can be under shadow making that part darker to easily deceive the state-of-the-art detection models. Leaves have some reflective properties which can make them over-exposed in photographs taken under bright sunlight, creating another obstacle in leaf detection.

To overcome all these challenges, we propose a novel method that helps improve detection performance with the help of shape priors. We use a clustering algorithm [9] to learn shape priors, which we use to help the model learn better to detect the object of interest from a small number of data points.

## 2 Related Works

Artificial intelligence (AI) and machine learning (ML) have found applications in various domains; agriculture is one of them. In agriculture, AI and ML are being used in precision farming, yield forecasting, weed and disease detection, etc. [10]. Vision-based systems are being extensively developed due to the very fast development of smartphones in the last decade, making them cheap as well as convenient.

The introduction of the Plant Village [4] dataset paves the way for plant disease detection to be framed reliably as an image-based pattern recognition problem. Mohanty et al. [1] for the first time use convolutional neural networks (CNN) for the classification of diseases in Plant Village. They use GoogleNet and AlexNet networks, which respectively reach 99.34% and 99.27% accuracy [1]. In consecutive work on the same dataset, they try to approach this problem in a different way [2]. Ahmed et al., in their work, make use of different image features like standard deviation and mean of different colour channels, entropy,



inverse difference, etc. for disease classification, reporting a 99.31% accuracy. As of now, an ensemble model has shown 100% classification accuracy for disease detection in the Plant Village dataset [4]. In [11], the authors use a histogram of oriented gradients (HOG) [13] along with features extracted by a CNN backbone for disease classification. This shows the possibility of feature fusion between learned deep representations and conventional image processing based features for better convergence of the network.

In a different line of work related to plant disease detection, researchers have framed the work as an object detection problem. Fuentes et al., in their works [6–8], have applied and improved different object detection algorithms for disease and pest detection in tomato plants. In [8], he experiments with different baseline algorithms, namely faster region-based convolutional neural networks (Faster-RCNN) [16], single-stage detectors (SSD) [19], and region-based fully convolutional networks (R-FCN). In the following works, he improves the detection performance with the help of a secondary diagnosis unit [6] and control classes [7] that improve the classification part of the detection. This approach performs better in real-time field scenarios.

Object detection is one of the fundamental tasks in computer vision. Earlier methods used handcrafted features like edge features, corner features, template matching, etc. These methods, though fast and lightweight, are unable to generalise to different scales and lighting. The Viola-Jones model [12] followed by the discovery of HOG features [13] successfully demonstrate the use case of machine learning techniques for object detection. In 2014, Girshick et al. [14] first used deep learning for object detection, revolutionising the field. In his work [14], Girshick proposes the region-based convolutional neural network (RCNN). RCNN uses selective search algorithm to propose potential bounding boxes. A CNN backbone generates deep-feature embeddings of the proposed regions, which are then used by class-specific SVM classifiers for the classification of the object. In the subsequent work, the author improves the speed and accuracy of the model by introducing the Fast-RCNN detector, which can generate feature embeddings for all the proposed regions together and uses a multi-layer perceptron for classification. The most widely used and latest model from the RCNN family is Faster-RCNN [16]. Ren et al. in [16] introduce region proposal networks (RPN), which are convolutional neural networks doing the same task that selective search does for RCNN but in a more precise manner. RPNs can learn to propose regions from the dataset, bringing down the number of proposals needed for accurate detection. It opens up a computation space for heavier classifiers down the line for the classification of the object. Single-stage detectors like you only look once (YOLO) [18] and single-shot multibox detectors (SSD) [19] were introduced later on for faster, real-time detection. These methods don't use region proposals like Faster-RCNN [16], which is a two-stage detector. The most recent work on object detection based on transformer architecture is detection transformer (DETR) [20]. DETR [20] uses a CNN backbone to generate features, which are then passed on to an encoder-decoder architecture. The decoder directly proposes a set of bounding boxes. The previous works

used a set of priors like anchors and anchor boxes, whereas DETR [20] is independent of priors and uses bipartite matching between targets and prediction. They introduce a novel Hungarian Loss [20] for bipartite matching, creating a paradigm shift in the field of object detection.

### 3 Our Method

#### 3.1 Generating shape priors

Suppose we have an object detection dataset,  $D$  with bounding boxes present in it. Suppose  $D'$  is a dataset containing the objects of interest present in  $D$  with only one object present in one image. In absence of a  $D'$ , it can be created by cropping the bounding boxes from  $D$ . This  $D'$  dataset will be used to learn shape prototypes to improve the object detection performance in  $D$ . Suppose the images present in  $D'$  are represented as  $I'_k$  where  $k \in 1, 2, \dots, N'$  where  $N'$  is the number of samples present in  $D'$ .

The images in  $D'$  are converted to grayscale to only retain shape features. These grayscale images have to be segmented with the help of a segmentation algorithm that can remove the background leaving only the object of interest in the image. We apply the ‘‘Deep Transformation-Invariant Clustering’’ [9] algorithm created by Monnier et al. to learn shape prototypes. Aim of the clustering algorithm is to generate  $n$  clusters of the dataset  $D'$ . A set of prototypes (mean of clusters)  $P_i$  are initialized randomly in the beginning. A set of parametric transformations  $\phi_{f_i(I'_k)}$  are chosen to be applied on the prototypes, where the parameter of the transformation,  $f_i(I'_k)$  is dependent on the sample with which we have to calculate the distance of the transformed prototype.  $f_i$  is a learnable function which takes the sample  $I'_k$  as input and generates parameters of the transformation  $\phi$ . The transformation aligns the prototype with the sample to accurately measure the distance with it so that the sample can be assigned to a cluster more accurately. The paper proposes a loss function of the form

$$L_{DTI}(P_{1:n}, f_{1:n}) = \sum_{i=1}^{N'} l(\phi_{f_1(I'_i)}(P_1), \dots, \phi_{f_n(I'_i)}(P_n)) \quad (1)$$

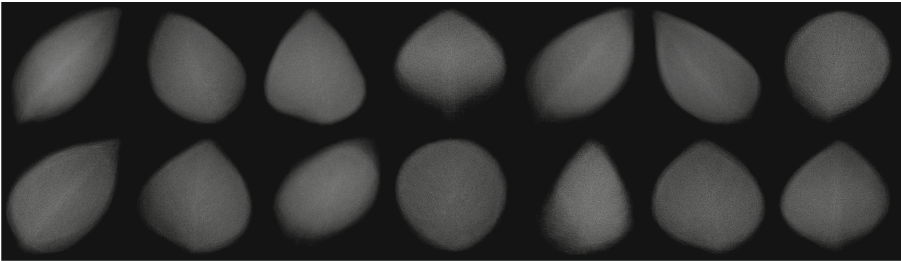
Here  $P_{1:n}$  and  $f_{1:n}$  represent the set of all the prototypes and the set of learnable functions associated with the prototypes respectively.  $l$  is the loss function of the corresponding clustering method. In case of the k-means DTI clustering, the loss takes a form

$$L_{DTIK-means}(P_{1:n}, f_{1:n}) = \sum_{i=1}^{N'} \min_u \|I'_i - \phi_{f_u(I'_i)}(P_u)\|^2 \quad (2)$$

If this loss is minimized iteratively with alternating optimization method for both  $P_i$ 's and the parameters of  $f_i$ 's, the model jointly learns the prototypes and the parameters of the parameter-prediction-networks ( $f_i$ 's). The iterative

optimization results in better cluster assignments which in-turn result in better prototype generation and vice-versa.

This method does prototype based clustering in pixel space for which the generated prototypes are visually interpretable. As this algorithm is invariant to a given set of parametric transformations (affine,color,spatial,morphological), it is suitable to learn prototypes from object instances of different scales, shape and orientation. DTI-Clustering algorithm learns deep parameter predictors ( $f_i$ ) which predict transformation parameters for each prototype. These parameters are used as inputs for transformation modules. Suppose we learn  $n$  number of prototypes from  $D'$  and  $P_i$  is the  $i$ 'th prototype where  $i \in 1, 2, 3, \dots, n$ .



**Fig. 2.** Fourteen of Our fifty prototypes that we generated from Plant Village [4] with the help of DTI Clustering algorithm [9]. (Best seen in digital version)

### 3.2 Prior matching

Template matching is one of the oldest methods in object detection. Let, we apply  $m$  number of transformations on each prototype,  $P_i$  which provides us with  $n \times m$  number of priors, where  $P_i^j$  is the  $i$ 'th protorype transformed by the  $j$ 'th method.  $I$  is an image from the dataset  $D$ . While detecting object in the image,  $I$  we convert it to grayscale image  $I_g$  and apply prior matching on it using  $P_i^j$  priors. This generates  $n \times m$  number of feature maps for us,  $R_i^j$  or  $r_i^j$ . We use normalized correlation coefficient and correlation coefficient method for prior matching [23].

The correlation coefficient between the prior  $P_i^j$  and region  $I_g(x, y)$  of the image  $I$  is given as,

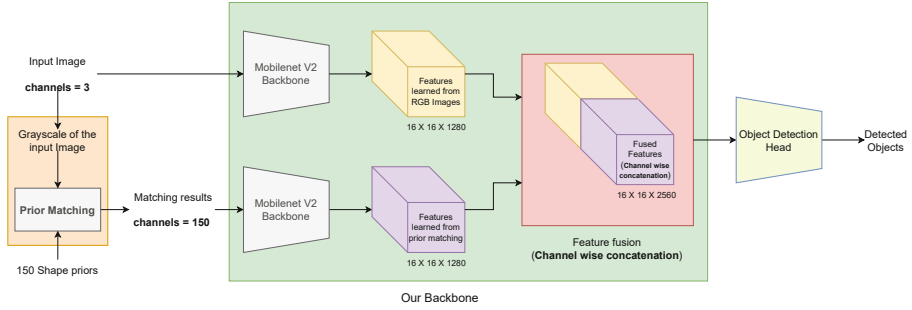
$$R(x, y) = \sum_{x',y'} (P_i^j(x', y') \cdot I_g(x + x', y + y')) \tag{3}$$

and the normalized cross correlation is given by,

$$r(x, y) = \frac{R(x, y) - \mu}{\sigma_{P_i^j} \cdot \sigma_{I_g}} \tag{4}$$

where  $\sigma_{P_i^j}$ ,  $\sigma_{I_g}$  and  $\mu$  are standard deviation of the prior, standard deviation of the image region  $I_g(x, y)$  and mean of the prior.

Given an image  $I_g$  of dimension  $H \times W$  and a prior  $P_i^j$  of dimension  $w \times h$  generates a matching result of dimension  $(W - w + 1) \times (H - h + 1)$  which is smaller than the image  $I$ .



**Fig. 3.** Architecture of our proposed model for feature fusion in object detection with a MobilenetV2 backbone and Faster-RCNN object detection head [16]

### 3.3 Feature fusion

The results generated with prior matching,  $r_i^j$  being smaller than the image in dimension is resized in the same dimension of the image,  $H \times W$ . There are a total of  $n \times m$  results generated from the prior matching. These results,  $r_i^j$  are stacked together creating a tensor of dimension  $H \times W \times (n \times m)$  where  $(n \times m)$  can be thought as the number of channels of an image.

Our method uses a backbone architecture which takes two inputs and generates feature maps. We use two same CNN architectures parallelly in the backbone. One takes the image of dimension  $H \times W \times 3$  as input and the other one takes  $H \times W \times (n \times m)$ . One of the architecture learns to summarise features from the RGB image whereas the other one learns to summarise the features generated from prior matching results. Feature maps generated from both these CNN architectures are of same spatial dimension. This helps to fuse the features together. As this is an object detection task, preserving spatial information in every stage is very important. For that reason we apply channel-wise concatenation for feature fusion.

### 3.4 Object detection

These fused features are then passed on to an object detection head for it to predict the boxes, scores and labels. Boxes are given as a tensor  $B$  of dimension  $N \times 4$  made up of vectors  $b_i$  of dimension  $1 \times 4$  in a format of  $[x_1, y_1, x_2, y_2]$

**Algorithm 1** Object detection with prior matching

---

```

1: Read RGB Image  $I$ 
2: Resize  $I$  to  $W \times H$ 
3: Convert  $I$  to grayscale  $I_g$ 
4: Initialize list  $MR = []$ 
5:  $n \leftarrow$  number of prototypes
6:  $m \leftarrow$  number of scales
7: for  $i \leftarrow 1$  to  $n$  do
8:   for  $j \leftarrow 1$  to  $m$  do
9:     prior is  $P_i^j$ 
10:     $R = \text{PriorMatching}(I_g, P_i^j)$ 
11:    resize  $R$  into  $W \times H$ 
12:    append  $R$  in  $MR$ 
13:   end for
14: end for
15: stack  $MR \leftarrow M_r \in R^{(n \times m) \times H \times W}$ 
16:  $F_{rgb} = \text{Backbone}_{RGB}(I)$ 
17:  $F_{mr} = \text{Backbone}_{MR}(M_r)$ 
18:  $F = \text{ChannelwiseConcatanation}(F_{rgb}, F_{mr})$ 
19:  $B, S, L = \text{DetectionHead}(F)$ 
20: return boxes  $B$ , scores  $S$ , labels  $L$ 

```

---

where  $i \in 1, 2, \dots, N$ .  $(x_1, y_1)$  is the top-left vertex and  $(x_2, y_2)$  is the bottom-right vertex of a bounding box  $b_i$ .  $N$  is the number of bounding boxes predicted by the model for image. The scores are given as a tensor  $S$  of dimension  $N \times 1$  where each element of the tensor  $s_i \in [0, 1]$ .  $s_i$  is a confidence score for the  $i$ 'th bounding box. The model predicts class labels as tensor  $L$  of dimension  $N \times 1$  where each element  $l_i \in 0, 1$ . Here label 0 represents background and label 1 represents object.

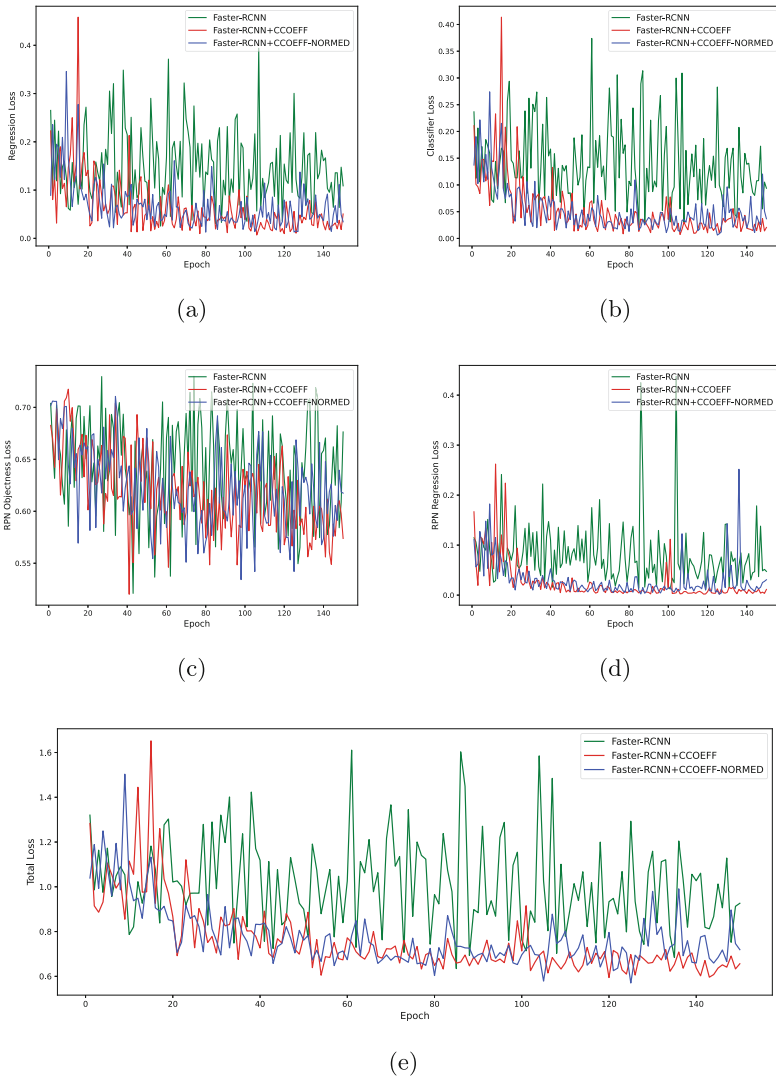
## 4 Experiments and Results

### 4.1 Experimental setup

We have used the PyTorch framework to implement this work on a system that has an RTX 3060 GPU with 12 GB of RAM. The system runs on an Intel i7-11700K processor and 16GB of RAM.

### 4.2 Dataset description

**Plant Village** [4] dataset consists of 61486 leaf images. There are 39 classes of leaves and background present in this class. The classes are plant-disease pairs. This dataset was collected in laboratory conditions, with one leaf in the image. Models trained on this dataset fail in real-time field conditions due to the presence of other leaves or objects.



**Fig. 4.** Comparison of (a) regression loss, (b) classification loss, (c) objectness loss of RPN, (d) regression loss of RPN and (e) total loss between our model and Faster-RCNN [16]

**Plant-Doc** [5] is also a plant disease detection dataset. There are two major advantages in using this dataset. This dataset was created by web scraping, and a major portion of the images are field images. Two, this dataset has an object detection dataset where leaves are the object of interest classified into 28 different plant-disease pairs. There are 8923 instances of leaves in a total of 2568

images. Training and testing data points are divided between 2355 and 243 in the dataset, and we use the testing dataset for validation.

### 4.3 Experimental setup

We have used the PyTorch framework to implement this work on a system that has an RTX 3060 GPU with 12 GB of RAM. The system runs on an Intel i7-11700K processor and 16GB of RAM.

### 4.4 Learning shape priors

Plant Village [4] dataset being clean is very suitable for learning prototypes. Leaf prototypes used as prior were learned from the Plant Village dataset. The images are first converted into grayscale with the help of the grayscale transformation present in the Pytorch library [22], and the background is removed to keep only the foreground leaf with the help of ‘rembg’ library [21]. This helps us segment the object of interest present in the foreground efficiently. As only the object of interest is present now, learning priors based on shape is easier for the model.

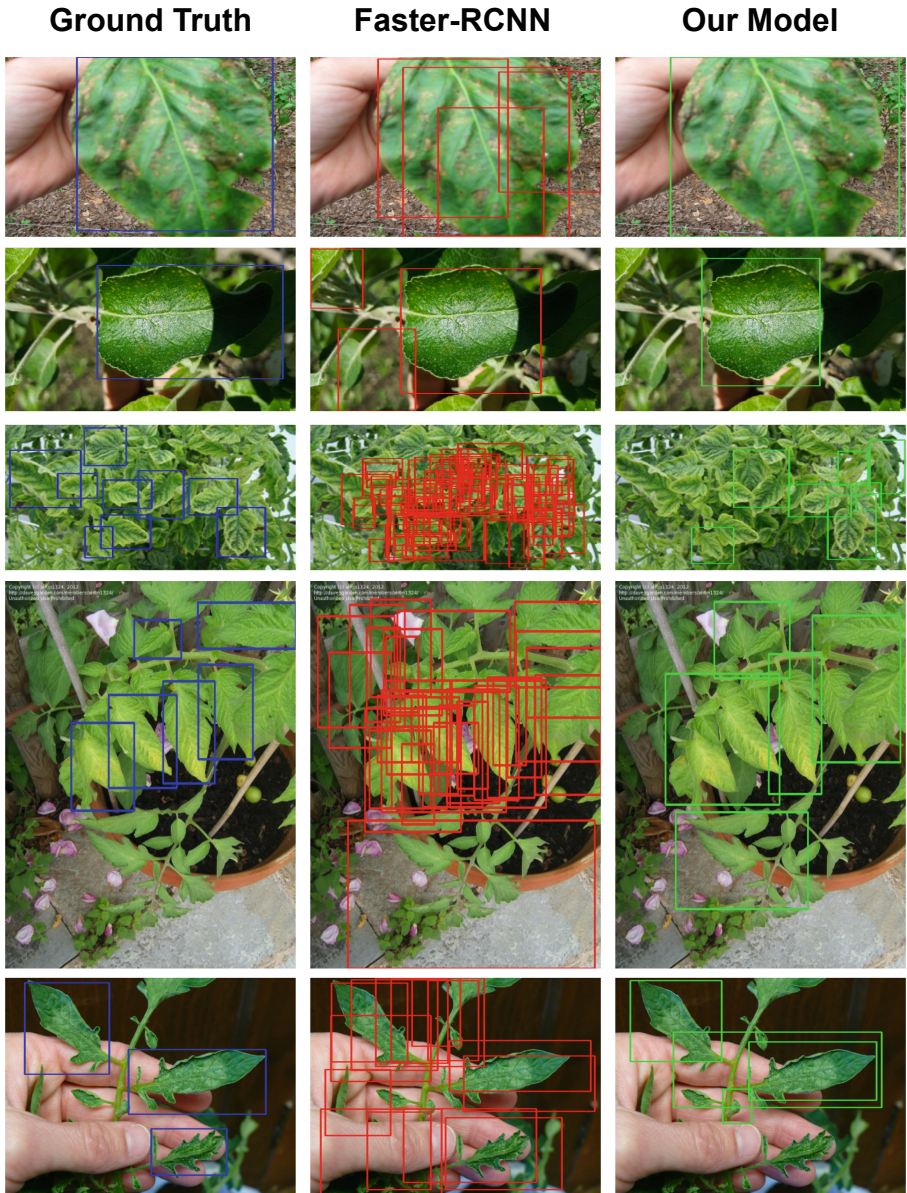
**Table 1.** Leaf detection performance of different state-of-the-art object detection models on Plant-Doc [5] dataset

Architecture	mAP(%)
YOLOv8 [28]	34.09
DETR [20]	8.61
Faster-RCNN [16]	46.8
Ours(Prior matching method: CCOEFF-NORMED [22])	<b>49.44</b>
Ours(Prior matching method: CCOEFF [22])	<b>48.69</b>

### 4.5 Leaf detection

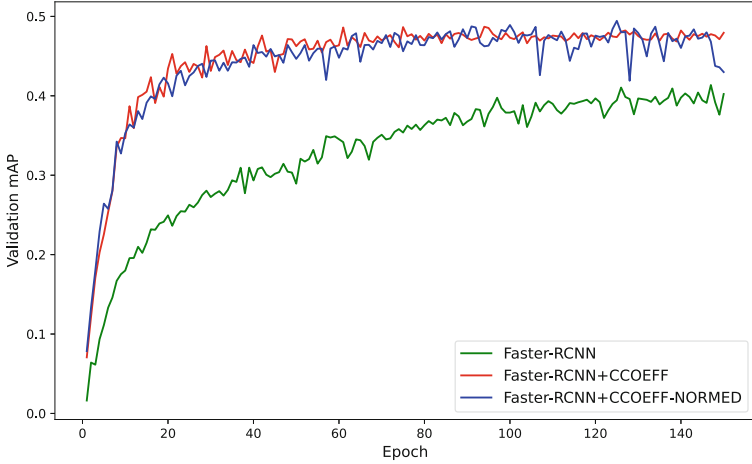
To train our object detection model, we use the stochastic gradient descent optimizer. The learning rate is initially 0.005. A learning rate scheduler with a step size of 20 epochs and a decay rate of 0.8 is employed for training. The optimizer has momentum and decay rates of 0.9 and 0.0005, respectively. There are only two classes: leaf and background.

We compare the performance of our model with different benchmark models of object detection. Our experiments are done on the Plant-Doc [5] dataset. In our experiments, the YOLOv8 [28] architecture reaches 34.09% mAP, Faster-RCNN [16] reaches 46.8% mAP, and DETR [20] converges at 8.61% mAP. Our model manages to reach 49.44% mAP, beating the best-performing model by **2.64%**.



**Fig. 5.** In the first column we can see the ground truth boxes drawn on some input images from the Plant-Doc [5] Object Detection dataset. In the second and third column predictions from baseline Faster-RCNN [16] and Our model are shown respectively





**Fig. 6.** Validation-mAP comparison of our model and Faster-RCNN [16]

The poor performance of DETR can be due to the absence of sufficient data for training as transformer architectures require huge amount of data to perform effectively [29].

These results show that our method is able to detect leaves with better precision than the existing state-of-the-art methods on leaf detection. In figure 5, we can see the predictions of Faster-RCNN [16] and our model side-by-side. In all the images, we can see that the baseline Faster-RCNN [16] model predicts a lot of false positive detections, especially if there are more leaves present in the input image, whereas our model predicts a smaller number of accurate predictions. A detection in the last row shows that our model is able to detect very small leaves, and if we look carefully, our model is also able to detect images from the backside.

The graphs in figure 4a, 4b, 4c, and 4d are plotted for four different losses used to train the Faster-RCNN [16] model for 150 epochs. In 4a, 4b, 4d, we can clearly see that the losses converge with increasing epochs very well for our model compared to the baseline Faster-RCNN [16]. The total loss plotted in figure 4e also supports the fact that our model converges better than the baseline Faster-RCNN [16] for leaf detection. The graph plotted in figure 6 shows that our model is able to converge a lot faster than the Faster-RCNN [16] model in a smaller number of epochs.

## 5 Ablation Studies

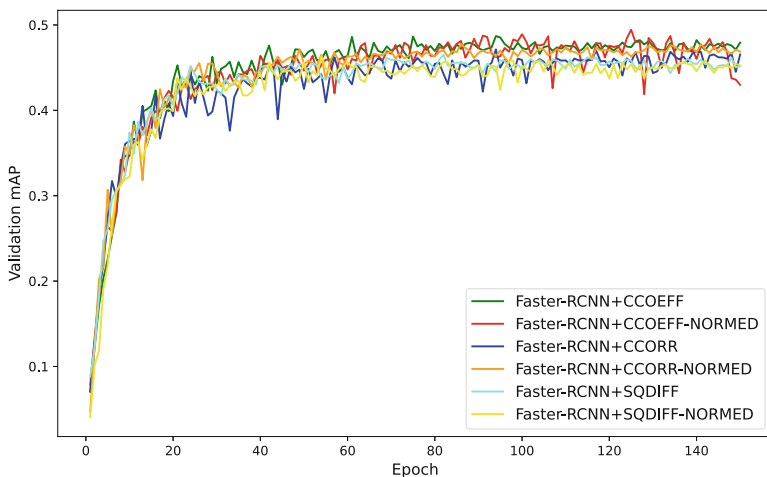
### 5.1 Prior matching

We use six different methods of prior matching present in the open cv [22] library to generate matching results which we use as input to the CNN backbone.

**Table 2.** Leaf detection performance of our method with different methods of prior matching [22]

Prior matching method	Method name	mAP(%)
Correlation coefficient [23]	CCOEFF	48.69
Normalised correlation coefficient [23]	CCOEFF-NORMED	<b>49.44</b>
Cross correlation coefficient [24]	CCORR	47.12
Normalised cross correlation coefficient [25]	CCORR-NORMED	47.76
Squared difference [26]	SQDIFF	46.51
Normalised squared difference [27]	SQDIFF-NORMED	46.33

Our experimental results showcased in table 2 and figure 7 show that using the normalised correlation coefficient(CCOEFF-NORMED [23]) implmented by the open cv library results into best performance of our model for leaf localization.

**Fig. 7.** Validation-mAP comparison for different prior matching methods used with our method

## 5.2 Prior matching with different scales of priors

We use unique combinations of three different scales of priors to study the dependence of our method on scales. We use three different scales of  $64 \times 64$ ,  $128 \times 128$  and  $256 \times 256$  for the resolution of the prior. We experiment taking each scale at a time. For priors of scale  $128 \times 128$  and  $256 \times 256$  we get detection mAP of 45.78% and 45.97% respectively. Whereas the performance for scale  $64 \times 64$  is comparatively low standing at 40.49%. When we use combination of two scales,

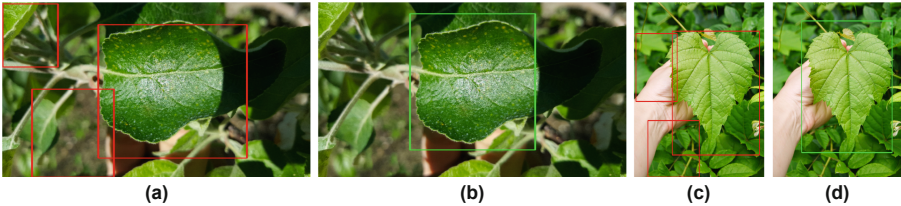
**Table 3.** Leaf detection performance of our method with different scales for prior matching

Scale of prior	mAP(%)
(64 × 64)	40.49
(128 × 128)	45.78
(256 × 256)	45.97
(64 × 64), (128 × 128)	43.33
(128 × 128), (256 × 256)	44.95
(64 × 64), (256 × 256)	44.53
(64 × 64), (128 × 128) and (256 × 256)	<b>49.44</b>

for using  $128 \times 128$  and  $256 \times 256$  we get the mAP of 44.95%. From table 3 we can see that our model outperforms the baseline only when all three scales are used together with a mAP of 49.44%.

### 5.3 Effect of shadows and sunlight

In figure 8 we can see that in the first two images, a part of the leaf is under shadow. Though this makes the job of the model harder, our model can be seen to be performing at par with the baseline Faster-RCNN. The images seen in the third and fourth images from left in figure 8 are of good sunlight conditions. In these conditions our model clearly out performs the baseline model.

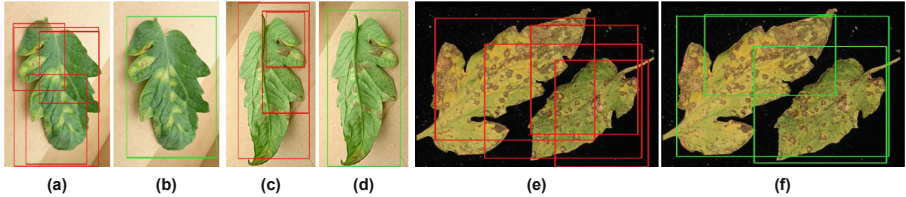


**Fig. 8.** Leaf detection performance of our model under shadows and sunlight conditions. (a), (c) are predictions from Faster-RCNN and (b), (d) are predictions from our model.

### 5.4 Compound leaf detection and false positives

In the examples of detection presented in figure 9 we can see that different parts of a single leaf are being predicted as different leaves. All of the leaves present in 9 are tomato leaves. Tomato leaves are compound leaves which are made up of multiple leaflets that resemble the shape of a leaf and get detected by the

model. From the results we can observe that our model is getting less confused by this phenomena than the baseline Faster-RCNN model, improving precision.



**Fig. 9.** We can see how the model can get confused as it returns different parts of single leaf as different leaves. (a), (c), (e) are predictions from Faster-RCNN and (b), (d), (f) are predictions from our model.

## 5.5 Backbone independence

Although all our experiments are conducted with a MobileNetV2 backbone we also experiment with ResNet18 [28] backbone in which we can see that the baseline Faster-RCNN performs with a mAP of 33.22% whereas our model performs with a mAP of **39.98%**. These results show that even with a ResNet18 backbone, our method has been able to improve the performance by 6.76% which indicates that our method is independent of the CNN backbone used for feature extraction.

## 6 Conclusion

From the experiments conducted, we can conclude that the shape information of an object can be successfully used with deep object detection models. Using shape priors is very helpful in cases where a smaller amount of data is present to improve precision. When handcrafted features for a specific task are used along with deep embeddings, the model converges and performs better, as some task-specific information makes the learning process easier for the model.

**Acknowledgements.** This work is supported by IIT Kharagpur AI4ICPS I Hub Foundation, a.k.a AI4ICPS under the aegis of DST.

## References

1. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, 2016. Available: [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fpls.2016.01419>

2. Ahmad, N., Asif, H.M.S., Saleem, G., Younus, M.U., Anwar, S., Anjum, M.R.: Leaf Image-Based Plant Disease Identification Using Color and Texture Features. *Wireless Pers. Commun.* **121**(2), 1139–1168 (2021). <https://doi.org/10.1007/s11277-021-09054-2>
3. A. Bruno et al., "Improving plant disease classification by adaptive minimal ensembling," *Frontiers in Artificial Intelligence*, vol. 5, 2022. Available: [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frai.2022.868926>
4. A. Pandian and G. Geetharamani, "Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network," *Mendeley Data*, V1, 201<https://doi.org/10.17632/tywbtsjrjv.1>
5. D. Singh et al., "PlantDoc: A Dataset for Visual Plant Disease Detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, Hyderabad India: ACM, Jan. 2020*, pp. 249–25<https://doi.org/10.1145/3371158.3371196>
6. A. F. Fuentes et al., "High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank," *Frontiers in Plant Science*, vol. 9, 2018. Available: [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fpls.2018.01162>
7. A. Fuentes et al., "Improving Accuracy of Tomato Plant Disease Diagnosis Based on Deep Learning With Explicit Control of Hidden Classes," *Frontiers in Plant Science*, vol. 12, 2021. Available: [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fpls.2021.682230>
8. A. Fuentes et al., "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors*, vol. 17, no. 9, 201<https://doi.org/10.3390/s17092022>
9. T. Monnier, T. Groueix, and M. Aubry, "Deep Transformation-Invariant Clustering," presented at *NeurIPS*, 2020
10. M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sensing of Environment*, vol. 236, p. 111402, Jan. 202<https://doi.org/10.1016/j.rse.2019.111402>
11. X. Fan et al., "Leaf image based plant disease identification using transfer learning and feature fusion," *Computers and Electronics in Agriculture*, vol. 196, p. 106892, May 202<https://doi.org/10.1016/j.compag.2022.106892>
12. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA: IEEE Comput. Soc, 2001*, p. I-511–I-51<https://doi.org/10.1109/CVPR.2001.990517>
13. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Jun. 2005, pp. 886–893 vol. <https://doi.org/10.1109/CVPR.2005.177>
14. R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation." *arXiv*, Oct. 22, 2014. Available: <http://arxiv.org/abs/1311.2524>
15. R. Girshick, "Fast R-CNN." *arXiv*, Sep. 27, 2015. Available: <http://arxiv.org/abs/1504.08083>
16. S. Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *arXiv*, Jan. 06, 2016. Available: <http://arxiv.org/abs/1506.01497>
17. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO (Version 8.0.0) [Computer software]," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>

18. J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection." arXiv, May 09, 2016. Available: <http://arxiv.org/abs/1506.02640>
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
20. N. Carion et al., "End-to-End Object Detection with Transformers." arXiv, May 28, 2020. Available: <http://arxiv.org/abs/2005.12872>
21. D. Gatis, "Rembg". Available: <https://github.com/danielgatis/rembg/tree/main>
22. G. Bradski, "The OpenCV Library", Dr. Dobb's Journal of Software Tools, 2000. Available: <https://github.com/opencv>
23. Wang, Z., Adelson, E.H.: Representing Moving Images with Layers. IEEE Trans. Image Process. **3**(5), 625–638 (1994)
24. Cross, G.: Automatic Similarity Detection in Digital Images. Pattern Recogn. **7**(3), 119–123 (1975)
25. Lewis, J.P.: Fast Normalized Cross-Correlation. Vision Interface **10**(1), 120–123 (1995)
26. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
27. K. Tieu and P. Viola, "Boosting Image Retrieval," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2000, pp. I-2 <https://doi.org/10.1109/CVPR.2000.854754>
28. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–77 <https://doi.org/10.1109/CVPR.2016.90>
29. A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, Jun. 03, 2021. Accessed: Dec. 28, 2023. [Online]. Available: <http://arxiv.org/abs/2010.11929>



# Spectral Aggregation Cross-Square Transformer for Hyperspectral Image Denoising

Yang Liu<sup>1</sup>, Yantao Ji<sup>1</sup>, Jiahua Xiao<sup>1</sup>, Yu Guo<sup>2</sup>, Peilin Jiang<sup>2</sup>,  
Haiwei Yang<sup>2</sup>, and Fei Wang<sup>2</sup>

<sup>1</sup> School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

{1228754216, jyt1262556482, xjh847286495}@stu.xjtu.edu.cn

<sup>2</sup> National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

{yu.guo, pljiang, wfx}@xjtu.edu.cn

**Abstract.** Hyperspectral image(HSI) denoising addresses noise impact during image acquisition. Transformers have gained notable prominence in the field of denoising, but their quadratic self-attention complexity poses computational challenges, hindering global information processing. Classical window-based self-attention limits non-local information flow, hampering large-scale object capture in HSI. Furthermore, spectral variations among neighboring bands introduce redundancy, which burdens the model and diminishes token variability, resulting in over-smoothing in the attention map. To address these issues, we propose a novel method, named Spectral Aggregation Cross-Square Transformer(SACT). We introduce a cross-square self-attention mechanism to enhance information exchange between windows, capturing long-range dependencies within spatial intra-spectrum from multiple perspectives. Spectrally, it extends the attention region horizontally, surrounding, and vertically, exploring omnidirectional spatial correlations among different receptive windows. Additionally, a spatial-spectral aggregation self-attention module is designed to capture global contextual dependencies across spatial and spectral dimensions, reducing spectral redundancy computation. Our method has evaluated synthetic and real hyperspectral datasets and shows SACT's effectiveness in enhancing both quantitative and qualitative HSI denoising performance.

**Keywords:** Hyperspectral image denoising · attention mechanism · spatial-spectral aggregation · vision transformer

---

Supported by National Major Science and Technology Projects of China (No.2019ZX01008101), Shaanxi Provincial Science and Technology Development Plan Project(2023-YBGY-033).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15315, pp. 458–474, 2025.  
[https://doi.org/10.1007/978-3-031-78354-8\\_29](https://doi.org/10.1007/978-3-031-78354-8_29)

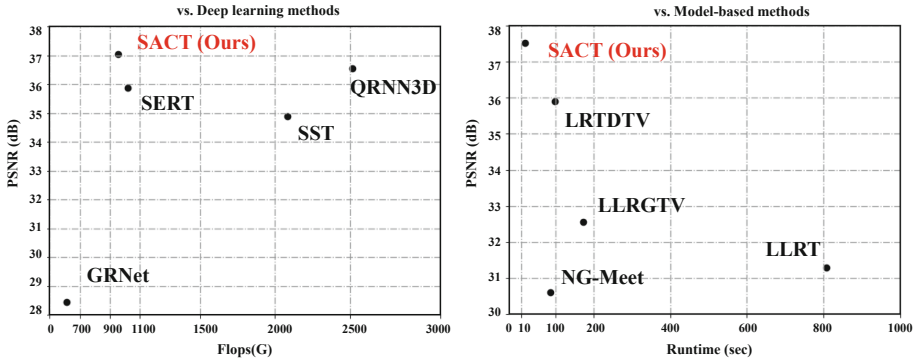


Fig. 1. Denoising performance *vs.* Flops(G) and Time(s) under CAVE dataset [1].

## 1 Introduction

Hyperspectral images represent a precious form of remote sensing data, offering the capability to capture surface feature information across a broader spectral range. HSI encompasses the spectral characteristics of objects as they reflect or emit light at different wavelengths, providing a wealth of information. Unlike traditional RGB and gray images, the unique spectral properties of HSI enable them to store more detailed and comprehensive information. Consequently, HSI finds applications in various fields, including remote sensing, object detection, agriculture, image processing, and more[2,3].

In recent years, model-based approaches have been the primary method of denoising HSI. These approaches combine signal processing techniques with mathematical models to identify the sources of noise in HSI. However, the application of model-based methods to real-world HSI denoising is hindered by the fact that real-world noise does not follow the assumptions of artificial mathematical models, presenting a significant challenge in practical scenarios.

In contrast to model-based methods, recently deep learning-based HSI denoising approaches primarily employ an end-to-end, data-driven methodology. These approaches directly learn noise patterns and features from noised HSI to achieve more precise denoising performance. Currently, there are two main DL architectures: Convolutional Neural Networks (CNNs) and Transformers. CNNs[4] show excellent performance, but are usually constrained by local filters when dealing with HSI, making it challenging to capture global spatial-spectral correlations. On the other hand, ViTs[5] are primarily designed to process spatial information, with limited consideration to exploit the distinctive spectral properties inherent in HSI. This characteristic leads to the computation of a significant volume of redundant information within the context of HSI.

To address the issues above, we propose a spectral aggregation cross-square Transformer, aimed at leveraging spatial-spectral information in HSI more effectively and efficiently to achieve comprehensive denoising, as shown in Figure 1.



First, we design a Cross-Square Self-Attention(CSSA). By incorporating triple-interacting receptive fields to facilitate inter-window information dissemination, CSSA can better capture spatial similarities. Specifically, CSSA partitions feature randomly into three distinct segments to construct cross-square attention regions, ensuring that information from different areas of HSI can be analyzed efficiently and comprehensively.

In addition, we develop a Spatial-Spectral Aggregation Self-Attention(SASA) to improve spatial-spectral context modeling further. SASA combines the neighboring spatial regions of adjacent bands into a single powerful token. This allows our network to capture the global contextual relationships between the spatial and spectral dimensions. Thanks to CSSA and SASA, our proposed SACT not only effectively exploits the intra-band long-range dependencies in the HSI, but also provides a meaningful tessellation, thus achieving a more comprehensive denoising effect.

Experimental denoising results on various datasets demonstrate that our SACT outperforms existing methods in terms of objective metrics on synthetic datasets and visual quality on real datasets. In summary, the primary contributions of our work are as follows:

- We propose a spectral aggregation cross-square Transformer for HSI denoising. SACT allows for a more comprehensive exploration of spatial non-local information and inter-band similarity in HSI.
- We propose Cross-Square attention and Spatial-Spectral Aggregation to enhance spatial information exploration, learn global HSI representations efficiently, and maximize denoising potential in the SACT network.
- We evaluate SACT on synthetic and real-world HSI for denoising. Our experiments show that our model outperforms other models both quantitatively and visually.

## 2 Related Work

### 2.1 Model-Driven-Based Methods

Traditional HSI denoising methods often rely on model-driven approaches, iteratively optimizing with manually crafted priors to restore clean HSI. For example, [6] estimates noise by considering pixel averages and structures, while [7] adaptively adjusts noise thresholds using a total variation to better utilize spatial-spectral correlation. Recent methods incorporate low-rank matrix approximation and tensor factorization concepts, such as combining spatial-spectral information with low-rank matrix decomposition and total variation regularization [8]. Similarly, we aggregate global spatial-spectral information by guiding the denoising network with tokens that aggregate more effective information.

### 2.2 Convolutional Neural Networks

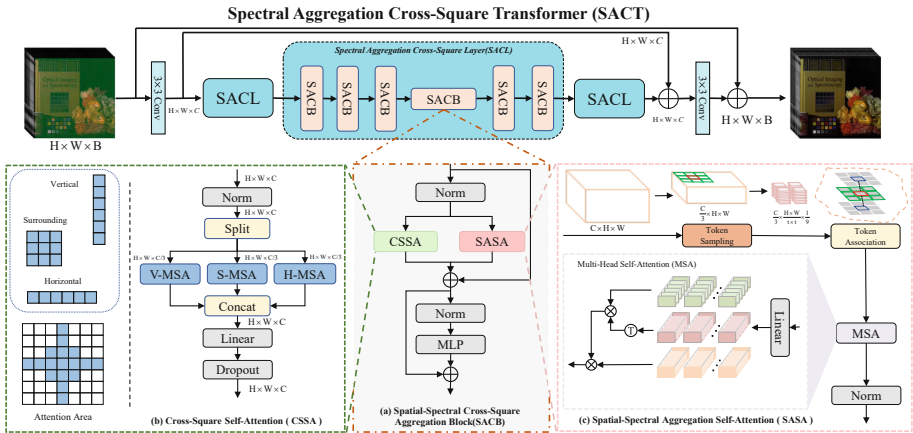
CNNs, unlike model-based approaches, automatically extract noise patterns without prior knowledge of noise types, garnering significant attention. Methods

like leverage 2D convolutions for denoising[4,9–12], demonstrate strong performance in spatial and spectral domains. Additionally, 3D convolution methods excel in capturing spatial-spectral structural information in high-dimensional data. QRNN3D effectively models spatial-spectral dependencies and global spectral correlations or introduces innovative regularization terms to mitigate HSI noise, enriching the landscape of HSI denoising approaches[4,13].

### 2.3 Vision Transformer

The Vision Transformer (ViT) has gained traction in computer vision recently by adopting the Transformer model, originally designed for natural language processing tasks, as a novel architecture [5,14]. ViT exhibits strong performance across various computer vision tasks, including image classification, object detection, and segmentation, owing to its ability to handle images of arbitrary sizes and demonstrate strong generalization capabilities[15–17].

In HSI denoising methods based on the transformer architecture, [18] enhances denoising capabilities with window attention modules and global spectral attention, focusing on spatial similarity and spectral dependencies. [19] employs rectangular window attention and extracts global-level low-rank characteristics of spatial-spectral cubes to suppress noise, exploring non-local spatial similarities and global spectral low-rank properties of HSI. Inspired by [20], we shift attention to higher-dimensional features, deploying the strategy across the entire global HSI to aggregate more powerful spatial-spectral tokens effectively.



**Fig. 2.** (a) The fundamental components, the spectral aggregation cross-square Block, primarily consist of CSSA and SASA. (b) CSSA, enhances spatial relationships within hyperspectral data by utilizing a cross-square attention window. (c) SASA, aggregates spatial-spectral information efficiently by power tokens.

### 3 Methods

#### 3.1 Overall Network Architecture

Our proposed spectral aggregation cross-square Transformer comprises three layers, illustrated in Figure 2 (a). Each layer starts with a  $3 \times 3$  convolutional operation for initial feature extraction from the input HSI. Subsequently, a spectral aggregation cross-square block is employed for spatial-spectral modeling, featuring two parallel branches: CSSA and SASA. More explanations of modules are provided in the supplementary material.

SACT integrates a robust self-attention mechanism for capturing long-range dependencies in the input spatial-spectral data. This module facilitates the network's understanding of non-local relationships among HSI regions, enhancing denoising efficacy. Initially, the network performs convolution operations on the input data:

$$F = K_1 \otimes Y \quad (1)$$

where  $Y$  represents the input HSI,  $K_1$  is a  $3 \times 3$  convolutional kernel,  $\otimes$  denotes the convolutional operation, and  $F$  represents the feature map. SACT is particularly effective in dealing with features at different scales, which is essential for denoising HSI. This is especially important since HSI usually contains multiple spectral bands with different spatial resolutions. The network comprises three Transformer layers, the structural process of each layer can be represented as:

$$\begin{aligned} F_1 &= \text{CSSA}(\text{LN}(F)) \\ F_2 &= \text{SASA}(\text{LN}(F)) \end{aligned} \quad (2)$$

The SACT model offers a straightforward structure, ease of implementation, and adjustable parameters, making it a valuable tool for HSI denoising. It comprises two modules: CSSA and SASA, responsible for denoising spatial and spectral features of HSI, respectively. This architecture effectively extracts valuable information from input data, yielding high-quality denoising outcomes. To mitigate gradient issues and enhance feature fusion, skip connections are employed in, simplifying network training and boosting performance. The denoised HSI can be represented as follows:

$$X_o = K_2 \otimes (F \oplus (F_1 + F_2)) \quad (3)$$

where  $X_o$  represents the denoised HSI,  $K_2$  is a  $3 \times 3$  convolutional kernel, and  $\oplus$  denotes the skip-connections operation.

#### 3.2 Cross-Square Self-Attention

For a given noisy HSI feature, it can be represented as  $F^{N \times H \times W \times C}$ .  $C$  is the number of channels. To better explain our model, we divide the cross-square windows into three categories: horizontal, surrounding, and vertical windows,

represented by  $h$ ,  $w$ , and  $m$ , respectively, illustrated in Figure 2 (b). Along the spectral dimension, we split it into three equal parts:

$$\begin{aligned} F_h, F_s, F_v &= \text{Split}(F) \\ F_o &= \text{Concat}(CSSA_h(F_h), CSSA_s(F_s), CSSA_v(F_v)) \end{aligned} \quad (4)$$

where  $F_o^{N \times H \times W \times C}$  denotes the output of CSSA layer,  $\{F_h, F_s, F_v\} \in \mathbb{R}^{N \times H \times W \times \frac{C}{3}}$  represents the features that are divided into three equal parts. These parts are then separately processed through horizontal, surrounding, and vertical CSSA layers, respectively.

Let's take the example of the vertical profile of CSSA. Given a rectangular region with a size of  $(h, w)$ , it is divided into  $n = \frac{H \times W}{h \times w}$  patches in the spatial dimension (where  $h < w$ ). In the spatial dimension,  $F^v$  is divided into  $\{F_1^v, F_2^v, \dots, F_n^v\}$ , where  $\{F_i^v\} \in \mathbb{R}^{h \times w \times \frac{C}{3}}$ . Then all the features  $F_i^v$  are passed through the Cross-Square Self Attention. Subsequently, each feature patch is linearly transformed as follows:

$$\begin{aligned} q_i, k_i, v_i &= \text{linear}(F_i^v) \\ Q_i &= F_i^v W_{q_i}, K_i = F_i^v W_{k_i}, V_i = F_i^v W_{v_i} \end{aligned} \quad (5)$$

where query  $Q_i$ , key  $K_i$ , value  $V_i$  are all  $\in \mathbb{R}^{h \times w \times \frac{C}{3}}$ ,  $W_{q_i}, W_{k_i}, W_{v_i}$  are learnable parameters  $\in \mathbb{R}^{\frac{C}{3} \times \frac{C}{3}}$ . Afterward,  $Q_i, K_i$ , and  $V_i$  are divided into  $h$  heads using the projection method:

$$\begin{aligned} Q_i &= \{Q_i^1, Q_i^2, Q_i^h\} \\ K_i &= \{K_i^1, K_i^2, K_i^h\} \\ V_i &= \{V_i^1, V_i^2, V_i^h\} \end{aligned} \quad (6)$$

Then, the profile self-attention for each head is computed as follows:

$$F_i^j = \text{SoftMax}\left(\frac{Q_i^j K_i^j}{\sqrt{d}} + B\right) V_i^j, j = 1, \dots, h \quad (7)$$

where  $B$  represents a learnable parameter incorporating position information. In summary, the output is primarily calculated as:

$$CSSA(F^v) = \text{Integration}(F_1^1, \dots, F_h^h) \quad (8)$$

Similar to the vertical profile self-attention branch mentioned above, the calculation process for the surrounding and horizontal self-attention branches is also consistent. The main difference lies in the sizes of the horizontal, surrounding, and vertical windows, which are  $[h, w]$ ,  $[m, m]$ , and  $[w, h]$ , respectively. The design of the Cross-Square approach enables us to utilize both local and non-local spatial similarity information effectively. The overlapping surrounding regions are assigned higher weights, thus taking advantage of the high information density inherent in HSI.

### 3.3 Spatial-Spectral Aggregation Self-Attention

For each SASA, the Spatial-Spectral Aggregation distance used for initialization is denoted as  $\lambda = 3$ . First, feature fusion is carried out in the spectral domain:

$$F^a = \text{Aggregate}(F^{i*3-1}, F^{i*3}, F^{i*3+1}), i = 1, 2, \dots, C//3 \quad (9)$$

where  $F^a$  represents the aggregated features in  $\mathbb{R}^{c \times H \times W}$ , with  $c = \frac{C}{3}$ , aggregating every three spectral bands together.

To reduce computational complexity while retaining essential features, as shown in the Figure 2 (c), we divide the features into tokens with different spatial strides of  $h$  and  $w$ , following the concept from [21],  $F_{ij}^k = \text{Split}(F^a)$ , where  $k = \{1, \dots, c\}$ ,  $i = \{1, \dots, \frac{H}{h}\}$ ,  $j = \{1, \dots, \frac{W}{w}\}$ . The spatial feature information is aggregated once again through a loop of 9 iterations:

$$\begin{aligned} Q_{ij}^k &= \text{SoftMax}\left(\frac{F_{ij}^k S_{ij}^{kT}}{\sqrt{d}}\right) \\ S_{ij} &= \text{SoftMax}(QK^T / \sqrt{d}) \end{aligned} \quad (10)$$

where  $S_{ij}$  represents the original region of the  $ij$ -th feature area before aggregation, with a size of  $9 \times \mathbb{R}^{h \times w}$ .  $d$  represents the step size of the loop,  $k$  represents the number of loops, ranging from 1 to 9.  $S_t$  represents the feature area after aggregation, as shown in Figure 2 (c).

Since  $S_t$  is the aggregated feature of adjacent regions in spatial-spectral, applying a self-attention mechanism as follows:

$$Q = S^{ij} W^q, K = S^{ij} W^k, V = S^{ij} W^v \quad (11)$$

By using a similar calculation method as in subsection 3.2, SASA enhances global contextual dependencies along the spatial and spectral dimensions.

$$\text{Attn}(S) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (12)$$

Finally, we obtain the output of a SCABlock by concatenating the results of CSSA and SASA by 3 and then reshape to  $F_{powt}^{N \times H \times W \times C}$ .

### 3.4 Complexity Analysis

In this module, a complexity comparison is conducted between our proposed SACT and the standard global spatial-spectral self-attention method.

**Global Spatial-Spectral Self-Attention:** The complexity of the standard global spatial-spectral self-attention can be represented as:

$$\Omega(G3SA) = \Omega(H^2 \times W^2 \times C) + \Omega(H \times W \times C^2) \quad (13)$$

**Cross-Square Self-Attention:** The formula for Cross-square Self-Attention can be expressed simply as:

$$CSSA(W) = Softmax(q(W)k(W)^T/\sqrt{d})V(W) \quad (14)$$

where  $CSSA(W)$  represents the attention map computed over the input features, and  $W$  denotes the input feature windows with sizes  $[h, w]$ ,  $[m, m]$ , and  $[w, h]$ . Since we randomly split the spectral dimension into three windows, the computational complexity of CSSA can be expressed as:

$$\Omega(CSSA) = \left( \frac{2 \times H^2 W^2 \times C}{h \times w \times 3} + \frac{H^2 W^2 \times C}{m^2 \times 3} \right) \quad (15)$$

**Spatial-Spectral Aggregation Self-Attention:** The formula for Spatial-Spectral Aggregation Self-Attention can be expressed as:

$$\Omega(SASA) = \left( \frac{H \times W \times C^2}{t^2 \times 9 \times 3} \right) \quad (16)$$

where  $t$  represents the spatial resolution size of segmented spatial-spectral tokens. The total computational complexity of our model can be expressed as:

$$\Omega(SACT) = \Omega(CSSA) + \Omega(SASA) \quad (17)$$

Since  $(h, w, t)$  are factors of  $(H, W)$ , where  $(H, W, C)$  are predefined parameters. Our method's computational complexity is lower than that of the global spatial-spectral attention mechanism for any  $\{h, w, t, m\} < \{H, W\}$ .

## 4 Experiments

In this section, we conduct experiments on both synthetic and real data. Synthetic data are corrupted with mixed noise. We comprehensively evaluate our proposed model and design experiments to analyze its effectiveness. More details of experiments and datasets are provided in the supplementary material.

### 4.1 Experimental Setup

**Training Datasets** To train our proposed SACT, we follow a training dataset configuration consistent with [22], randomly selecting 100 images from the ICVL dataset [23]. After central cropping, each processed image has a spatial resolution of  $512 \times 512$  and 31 spectral bands. Subsequently, through operations like random cropping, rotation, and flipping, the spatial resolution of the images is adjusted to  $64 \times 64$ , resulting in approximately 53,000 training samples.

**Testing Datasets** For the synthetic dataset, our experiments include the ICVL, CAVE[1], and KAIST[24] datasets. In the ICVL test set, we randomly select 50 different images not included in the training set for testing. In the CAVE dataset, 30 images are chosen for testing, each with a spatial resolution of  $512 \times 512$  and 31 spectral bands. For the KAIST dataset, we select 10 images, and after cropping, each image has a spatial resolution of  $2048 \times 2048$  and 31 spectral bands. To ensure fair testing, we apply normalization uniformly to the test set.

To assess the model’s generalization, testing datasets also include remote sensing datasets, such as the Washington DC dataset<sup>1</sup>, Pavia University [25], and real datasets: Urban [26] and Realistic dataset [27].

**Evaluation Metrics and Implementation Details** We use three image quality evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM)[28], and Spectral Angle Mapper (SAM)[29].

**Table 1.** Quantitative comparison on synthetic, remote sensing, and real datasets. Best in **bold**, second best underlined.

Datasets + Noise	Metric	Noisy	LRTDTV [30]	LLRGTV [31]	GRNet [32]	MAC-Net [33]	T3SC [34]	SST [18]	SERT [19]	SACT
Kaist:N-Gaussian	PSNR	18.26	37.33	36.66	30.58	34.33	<u>37.79</u>	36.79	37.77	<b>38.96</b>
	SSIM	0.6145	0.9510	0.9109	0.9131	0.9537	<b>0.9915</b>	0.9774	0.9851	<u>0.9873</u>
	SAM	1.0741	0.2195	0.3678	0.5235	0.4795	<b>0.1603</b>	0.2267	0.1918	<u>0.1677</u>
Kaist:G+Stripe	PSNR	18.10	37.16	36.68	30.41	33.37	<u>37.68</u>	36.59	37.57	<b>38.95</b>
	SSIM	0.6136	0.9498	0.9152	0.9093	0.9520	<b>0.9915</b>	0.9779	0.9849	<u>0.9877</u>
	SAM	1.0746	0.2113	0.3510	0.5298	0.4733	<b>0.1678</b>	0.2294	0.1956	<u>0.1724</u>
Kaist:G+Deadline	PSNR	17.23	36.49	34.76	30.22	32.82	37.04	36.34	<u>37.35</u>	<b>38.57</b>
	SSIM	0.5770	0.9413	0.8761	0.9044	0.9428	<b>0.9903</b>	0.9762	0.9836	<u>0.9858</u>
	SAM	1.0960	0.2354	0.4268	0.5280	0.4718	<u>0.1888</u>	0.2323	0.1976	<b>0.1788</b>
Kaist:G+Impulse	PSNR	15.00	37.03	33.61	28.77	31.02	34.87	35.62	<u>36.58</u>	<b>38.04</b>
	SSIM	0.4625	0.9505	0.8502	0.8870	0.9205	<u>0.9808</u>	0.9726	0.9803	<b>0.9844</b>
	SAM	1.0965	0.2445	0.5109	0.5996	0.5925	0.2561	0.2622	<u>0.2262</u>	<b>0.1944</b>
Kaist:Mixture	PSNR	13.70	35.60	32.14	28.29	28.41	33.62	34.65	<u>35.71</u>	<b>36.84</b>
	SSIM	0.4216	0.9355	0.8166	0.8847	0.9041	0.9750	0.9638	<u>0.9761</u>	<b>0.9787</b>
	SAM	1.1133	0.2503	0.5218	0.5780	0.5840	0.2958	0.3050	<u>0.2459</u>	<b>0.2270</b>
ICVL:Mixture	PSNR	13.97	34.46	31.39	31.67	30.75	35.68	39.58	<u>40.44</u>	<b>41.68</b>
	SSIM	0.3392	0.9184	0.8756	0.9557	0.9332	0.9790	0.9928	<u>0.9940</u>	<b>0.9955</b>
	SAM	0.8987	0.1127	0.2538	0.1431	0.2673	0.1389	0.0480	<u>0.0470</u>	<b>0.0439</b>
CAVE:Mixture	PSNR	14.17	33.82	27.12	28.44	28.53	33.61	34.89	<u>35.86</u>	<b>37.03</b>
	SSIM	0.4188	0.9085	0.7221	0.8899	0.8920	0.9728	0.9616	<u>0.9737</u>	<b>0.9785</b>
	SAM	1.1371	0.2938	0.6567	0.6329	0.6234	0.4137	0.4095	<u>0.3403</u>	<b>0.3096</b>
PAVIA:Mixture	PSNR	13.89	29.65	28.41	26.57	27.34	31.39	32.85	<u>33.28</u>	<b>34.20</b>
	SSIM	0.3400	0.8963	0.8923	0.8536	0.8813	0.9523	0.9588	<u>0.9653</u>	<b>0.9700</b>
	SAM	0.9746	0.2445	0.3142	0.2815	0.3530	0.2314	0.1555	<u>0.1451</u>	<b>0.1429</b>
WDC:Mixture	PSNR	13.98	36.33	34.32	25.14	30.74	30.49	30.59	<u>31.33</u>	<b>37.20</b>
	SSIM	0.2148	0.8597	0.8260	0.7682	0.7740	0.9064	0.8924	<u>0.9098</u>	<b>0.9296</b>
	SAM	1.0120	0.2148	0.3150	0.3255	0.5371	0.1972	0.1860	<u>0.1609</u>	<b>0.1368</b>
Real:Mixture	PSNR	22.53	26.58	26.14	26.80	24.25	26.72	<u>26.95</u>	26.51	<b>27.54</b>
	SSIM	0.6147	0.8541	0.8558	0.8987	0.8224	0.9100	<u>0.9118</u>	0.9027	<b>0.9140</b>
	SAM	0.6147	<u>0.0620</u>	0.0626	0.1168	0.1480	<b>0.0576</b>	0.0711	0.0850	0.0676

<sup>1</sup> <https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html>

We randomly add the complex noise during network training to enhance the network’s generalization and robustness. Our training strategy is inspired by [22]. The learning rate is set to 1e-4. After 70 epochs, the learning rate is reduced by a factor of 10. The batch size for our proposed network is set to 8, and training is conducted on a Quadro RTX 8000 for 80 epochs.

## 4.2 Efficiency analysis

In this subsection, we analyze the performance of different methods regarding model parameter size and time consumption cost on the CAVE dataset with noise case 5. All test experiments are conducted on a 3090 GPU, and the results are presented in Table 2.

SACT demonstrates superior performance when using the Transformer-based architecture, achieving a great balance between low computational complexity and a reasonable number of parameters. Compared to other deep learning-based techniques, it also has faster computation times, demonstrating its remarkable balance between resource efficiency and performance.

## 4.3 Experimental Results

To ensure the effectiveness of the comparative methods, for model-based approaches, we select corresponding test settings, and for deep learning-based methods, we compare them with their publicly available pre-trained models. For datasets with a larger number of spectral bands, some comparative methods are only applicable to data with 31 spectral bands. We employ a sliding window strategy for these methods (GRNet, T3SC, SST, and SERT) for denoising, and the average of the results within all windows is used as the final result.

**Table 2.** Model complexity comparisons, parameter count (M), floating-point operations per second (FLOPS), and execution time (seconds), using the CAVE dataset with dimensions of 512×512×31.

Methods	QRNN3D [22]	GRNet [32]	MAC-Net [33]	T3SC [34]	SST [18]	SERT [19]	SACT
PSNR	36.55	28.44	28.53	33.61	34.89	35.86	37.03
Params	0.86	41.44	0.43	0.83	4.10	1.91	2.42
GFLOPS	2513.7	610.7	-	-	2082.4	1018.9	957.0
Times	0.720	0.466	2.709	0.758	2.265	0.872	0.821

**Complex Noise on Synthetic Datasets** To assess denoising performance under mixed noise settings, we conduct comparative evaluations on datasets ICVL, CAVE, KAIST, PAVIA, WDC, and Real. The upper half of Table 1 presents quantization and noise reduction performance on the KAIST dataset



under Complex Noise. Notably, SACT achieves a minimum PSNR improvement of 1.13 dB compared to SERT across various experiments adding case complex noise to the KAIST dataset. Figure 3 illustrates the visual comparison results under Case 5 on the KAIST dataset. In particular, we observe that GRNet exhibits a spectral shift phenomenon during the denoising process, with some pixels changing from yellow to blue. Similarly, T3SC also experiences problems with color changes. In contrast, SACT maintains better spectral consistency and visual fidelity closer to GroundTruth in terms of both spectral and spatial.

The lower part of Table 1 highlights SACT’s exceptional performance on other synthetic datasets with Gaussian+Mixture noise, suggesting its superiority in handling images with a wider spectral range. LRTDTV, LLRGTV, and MACNet exhibit some degree of pseudo-artifacts. In contrast, our proposed SACT method effectively removes Gaussian noise while preserving the details in the HSI more finely compared to other methods.

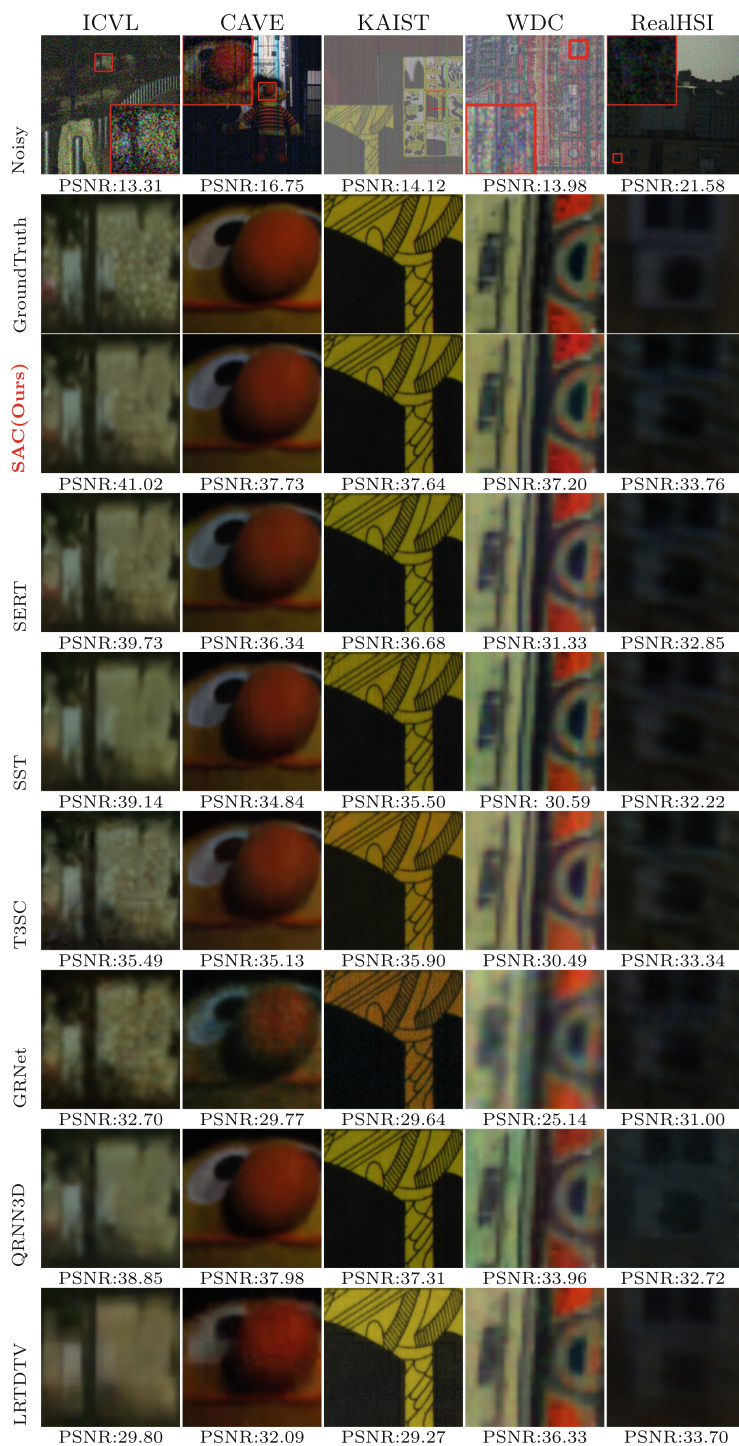
**Table 3.** Classification quantitative (OA) and qualitative results (Kappa) of Indian Pines dataset. Best in **bold**, second best underlined.

Metric	Noisy	LRTDTV	LLRGTV	GRNet	MACNet	T3SC	QRNN3D	SST	SERT	SACT(Ours)
OA(%)	76.30	88.45	81.75	77.19	76.14	84.29	<u>91.63</u>	84.38	89.59	<b>92.23</b>
Kappa	0.7216	0.7627	0.7542	0.7308	0.7194	0.8164	<u>0.9030</u>	0.8182	0.8793	<b>0.9097</b>

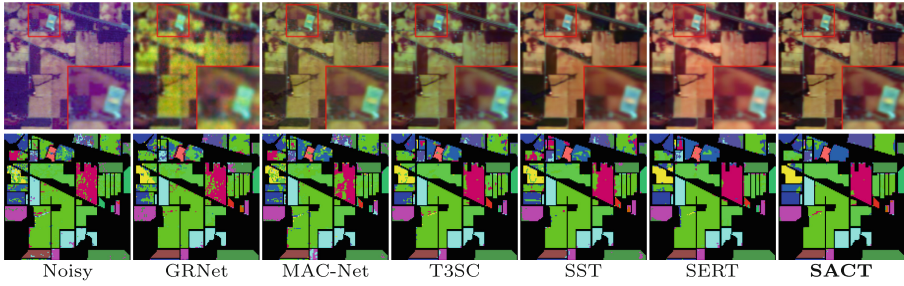
To validate the model’s performance under real-world noise conditions, we conducted hyperspectral image denoising experiments on the Urban, Indian Pines, and Realistic datasets, widely recognized as representative datasets for real-world scenarios.

Figure 3 in the far right column illustrates the visual performance of various methods on real datasets. The best PSNR and SSIM values indicate that our proposed SACT excels in real noise removal. The priori global low-rank assumptions of LRTDTV and LLRGTV align well with the SAM metric changes observed in denoising on real datasets, thereby mitigating the impact of noise. However, their performance is poor on additional WDC and other datasets. In contrast, our advantage lies in performing better across broader datasets, enhancing the model’s potential to adapt to various HSI denoising tasks. Our proposed SACT achieved a significant improvement of 0.59 dB in PSNR than others.

To demonstrate the effectiveness and practical significance of hyperspectral denoising for downstream tasks, we conducted classification experiments on the Indian Pines using denoised HSI produced by the aforementioned methods. Employing an SVM-based strategy with a 50% data split for training, we calculated classification accuracy metrics, including Overall Accuracy (OA) and the Kappa coefficient (Kappa). The results, as presented in Table 3, indicate that our proposed SACT achieved the highest classification performance, with an OA



**Fig. 3.** Visual comparisons on the CAVE datasets under Non-i.i.d Gaussian+Mixture noise.



**Fig. 4.** Visual and classification comparison of denoising on the Indian Pines. The HSI bands 127, 24, and 10 combine to create the visual pseudocolor image.

**Table 4.** Ablation study of CSSA and SASA using base function.

**Table 5.** Ablation study of different spectral enhancement strategies.

Baseline	CSSA	SASA	Params(M)	PSNR(dB)	SSIM	SAM
✓			1.73	33.47	0.8538	0.2311
✓	✓		1.75	36.26	0.9252	0.1374
✓		✓	2.40	34.42	0.8752	0.2201
✓	✓	✓	2.42	<b>37.20</b>	<b>0.9296</b>	<b>0.1368</b>

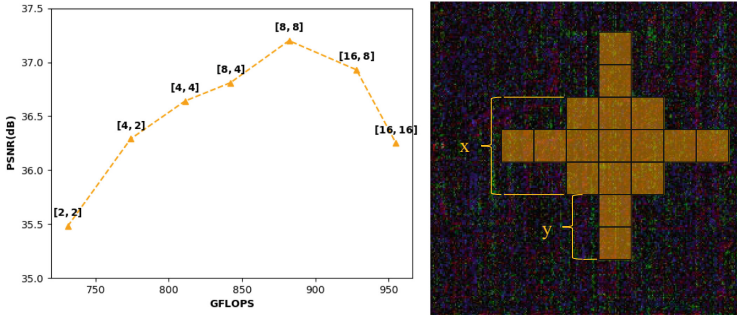
Method	PSNR(dB)	SSIM	SAM
Global SE[18]	35.61	0.9112	0.1555
Low-rank SE[19]	36.70	0.9201	0.1559
Spectral-aggregation SE(Ours)	<b>37.20</b>	<b>0.9296</b>	<b>0.1368</b>

of 92.23% and a Kappa coefficient of 0.9097. SACT is capable of recovering more denoising details in complex HSI while maintaining spectral consistency.

#### 4.4 Ablation experiment

**Module Effectiveness Analysis** In SACT, we design two modules to extract spatial-spectral similarity in HSI, named Cross-Square Self-Attention (CSSA) and Spatial-Spectral Aggregation Self-Attention (SASA). We use global feature extraction, shallow feature capture, and multi-layer perceptron modules as baselines to validate the effectiveness of spatial similarity utilization and spatial-spectral information fusion. The baseline incorporating the Cross-Square Self-Attention module is denoted as baseline + C, while the one incorporating the Spatial-Spectral Aggregation Self-Attention module is denoted as baseline + S. The results of the validation are presented in Table 4.

**Comparison of the effectiveness of different attention window mechanisms and spectral enhancement strategies** We evaluate spectral enhancement strategies, including global spectral attention and low-rank memory spectral enhancement. Table 5 presents the quantitative denoising results, allowing for a comprehensive comparison.



**Fig. 5.** Performance of different Cross-Square receptive fields on the WDC Mall dataset. In  $[x, y]$ ,  $x$  represents the size of the square, and  $y$  represents the distance of the cross (starting from the square’s edge).

#### 4.5 Parameter analysis

We explore the impact of different cross-square attention region sizes and spectral enhancement levels on model performance in Figure 5. Increasing window size elevates computational demands, initially boosting quantitative performance before plateauing. Balancing computational cost and effectiveness, we opt for a receptive field with a square edge length and cross-distance both set to 8.

### 5 Conclusion

In this paper, we present a spectral aggregation cross-square Transformer for HSI denoising. Our approach takes advantage of the intra-band correlations and dependencies within HSI, exploring spatial similarity more thoroughly for a more accurate spatial restoration. Additionally, we aggregate contextual information across the entire spatial-spectral domain to learn global HSI representations efficiently. The proposed modules generate fewer and more powerful tokens for self-attention operation, resulting in fewer parameters and lower computational complexity. Experiments on various datasets demonstrate the effectiveness and superiority of our approach.

**Acknowledgements.** We thank Qiuguang Zhang and Mengyuan Lin for their invaluable assistance in data processing and insightful discussions.

### References

1. Jong-Il Park, Moon-Hyun Lee, Michael D Grossberg, and Shree K Nayar. Multi-spectral imaging using multiplexed illumination. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007

2. Wang, Y., Li, D., Hanjie, W., Li, X., Kong, F., Wang, Q.: Multiple spectral-spatial representation based on tensor decomposition for hsi anomaly detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**, 3539–3551 (2022)
3. Priyanka Sahu, Amit Prakash Singh, Anuradha Chug, and Dinesh Singh. A systematic literature review of machine learning techniques deployed in agriculture: A case study of banana crop. *IEEE Access*, 10:87333–87360, 2022
4. Juntao Guan, Rui Lai, Huanan Li, Yintang Yang, and Lin Gu. Dnrcnn: Deep recurrent convolutional neural network for hsi destriping. *IEEE Transactions on Neural Networks and Learning Systems*, 2022
5. Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022
6. Ping, X., Chen, B., Xue, L., Zhang, J., Zhu, L., Duan, H.: A new mnf-bm4d denoising algorithm based on guided filtering for hyperspectral images. *ISA Trans.* **92**, 315–324 (2019)
7. He, W., Zhang, H., Shen, H., Zhang, L.: Hyperspectral image denoising using local low-rank matrix recovery and global spatial-spectral total variation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **11**(3), 713–729 (2018)
8. Sun, Y., Huang, J., Zhao, L., Kai, H.: Hyperspectral snapshot compressive imaging with dense back-projection joint attention network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**, 6099–6109 (2022)
9. Wei, X., Xiao, J., Gong, Y.: Blind hyperspectral image denoising with degradation information learning. *Remote Sensing* **15**(2), 490 (2023)
10. Jiahua Xiao, Yantao Ji, and Xing Wei. Hyperspectral image denoising with spectrum alignment. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5495–5503, 2023
11. Jiahua Xiao and Xing Wei. Hyperspectral image denoising using uncertainty-aware adjustor. In *IJCAI*, pages 1560–1568, 2023
12. Jiahua Xiao, Yang Liu, Shizhou Zhang, and Xing Wei. Bridging fourier and spatial-spectral domains for hyperspectral image denoising. In *ACM Multimedia*, 2024
13. Qiang Zhang, Yaming Zheng, Qiangqiang Yuan, Meiping Song, Haoyang Yu, and Yi Xiao. Hyperspectral image denoising: From model-driven, data-driven, to model-data-driven. *IEEE Transactions on Neural Networks and Learning Systems*, 2023
14. Swalpa Kumar Roy, Ankur Deria, Chiranjibi Shah, Juan M Haut, Qian Du, and Antonio Plaza. Spectral–spatial morphological attention transformer for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023
15. He, W., Huang, W., Liao, S., Zhen, X., Yan, J.: Csit: A multiscale vision transformer for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **15**, 9266–9277 (2022)
16. Mathias Gehrig and Davide Scaramuzza. Recurrent vision transformers for object detection with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13884–13893, 2023
17. Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11144–11154, 2023

18. Li, M., Ying, F., Zhang, Y.: Spatial-spectral transformer for hyperspectral image denoising. In Proceedings of the AAAI Conference on Artificial Intelligence **37**, 1368–1376 (2023)
19. Miaoyu Li, Ji Liu, Ying Fu, Yulun Zhang, and Dejing Dou. Spectral enhanced rectangle transformer for hyperspectral image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5805–5814, 2023
20. Yuanhao Cai, Jing Lin, Xiaowan Hu, Haoqian Wang, Xin Yuan, Yulun Zhang, Radu Timofte, and Luc Van Gool. Coarse-to-fine sparse transformer for hyperspectral image reconstruction. In *European Conference on Computer Vision*, pages 686–704. Springer, 2022
21. Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. Vision transformer with super token sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22690–22699, 2023
22. Wei, K., Ying, F., Huang, H.: 3-d quasi-recurrent neural network for hyperspectral image denoising. *IEEE transactions on neural networks and learning systems* **32**(1), 363–375 (2020)
23. Arad, B., Ben-Shahar, O.: Sparse Recovery of Hyperspectral Signal from Natural RGB Images. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9911, pp. 19–34. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46478-7\\_2](https://doi.org/10.1007/978-3-319-46478-7_2)
24. Choi, I.: MH Kim, D Gutierrez, DS Jeon, and G Nam. High-quality hyperspectral reconstruction using a spectral prior, Technical report (2017)
25. Paolo Gamba. A collection of data for urban area characterization. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 1. IEEE, 2004
26. Volodymyr Mnih and Geoffrey E Hinton. Learning to detect roads in high-resolution aerial images. In *European conference on computer vision*, pages 210–223. Springer, 2010
27. Tao Zhang, Ying Fu, and Cheng Li. Hyperspectral image denoising with realistic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2248–2257, October 2021
28. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004
29. Roberta H Yuhas, Joseph W Boardman, and Alexander FH Goetz. Determination of semi-arid landscape endmembers and seasonal trends using convex geometry spectral unmixing techniques. In *JPL, Summaries of the 4th Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1993
30. Wang, Y., Peng, J., Zhao, Q., Leung, Y., Zhao, X.-L., Meng, D.: Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **11**(4), 1227–1243 (2017)
31. He, W., Zhang, H., Shen, H., Zhang, L.: Hyperspectral image denoising using local low-rank matrix recovery and global spatial-spectral total variation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **11**(3), 713–729 (2018)
32. Cao, X., Xueyang, F., Chen, X., Meng, D.: Deep spatial-spectral global reasoning network for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2021)

33. Xiong, F., Zhou, J., Zhao, Q., Jianfeng, L., Qian, Y.: Mac-net: Model-aided nonlocal neural network for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2021)
34. Bodrito, T., Zouaoui, A., Chanussot, J., Mairal, J.: A trainable spectral-spatial sparse coding model for hyperspectral image restoration. *Adv. Neural. Inf. Process. Syst.* **34**, 5430–5442 (2021)



# SDformerFlow: Spiking Neural Network Transformer for Event-based Optical Flow

Yi Tian<sup>(✉)</sup> and Juan Andrade-Cetto

Institut de Robòtica e Informàtica Industrial, CSIC-UPC, Barcelona, Spain  
{ytian, cetto}@iri.upc.edu

**Abstract.** Event cameras produce asynchronous and sparse event streams capturing changes in light intensity. Overcoming limitations of conventional frame-based cameras, such as low dynamic range and data rate, event cameras prove advantageous, particularly in scenarios with fast motion or challenging illumination conditions. Leveraging similar asynchronous and sparse characteristics, Spiking Neural Networks (SNNs) emerge as natural counterparts for processing event camera data.

Recent advancements in Visual Transformer architectures have demonstrated enhanced performance in both Artificial Neural Networks (ANNs) and SNNs across various computer vision tasks. Motivated by the potential of transformers and spikeformers, we propose two solutions for fast and robust optical flow estimation: STTFlowNet and SDformerFlow. STTFlowNet adopts a U-shaped ANN architecture with spatiotemporal Swin transformer encoders, while SDformerFlow presents its full spike counterpart with spike-driven Swin transformer encoders.

Notably, our work marks the first utilization of spikeformer for dense optical flow estimation. We conduct end-to-end training for both models using supervised learning on the DSEC-flow Dataset. Our results indicate comparable performance with state-of-the-art SNNs and significant improvement in power consumption compared to the best-performing ANNs for the same task.

Our code will be open-sourced at <https://github.com/yitian97/SDformerFlow>.

**Keywords:** Spiking Neural Network · Event camera · Transformer · Optical flow

## 1 Introduction

Event cameras have a higher temporal resolution compared to traditional cameras, capturing per-pixel intensity changes. The sparse and asynchronous event streams they produce can directly encode apparent motion patterns (optical flow), enabling accurate motion estimation in challenging scenarios like fast motion or low illumination. However, due to the fundamentally different data throughput of the two camera types, estimating event-based optical flow suggests different approaches than conventional computer vision methods. Recent



research utilizing Artificial Neural Networks (ANNs) has demonstrated higher accuracy [10, 19] compared to model-based methods [26, 28] for event-based optical flow estimation. ANN layers rely on floating-point calculations and may not fully exploit the sparse and asynchronous nature of event data. Spiking Neural Networks (SNNs) have emerged as a promising match for event data. In SNNs, neurons integrate input spike trains and generate a binary spike when the membrane potential reaches a threshold, resetting its value afterward. Neurons are active only when spikes arrive, just as individual event camera pixels are active only when intensity changes. Sharing this event-driven characteristic makes SNNs an energy-efficient option for processing event data. However, directly training deep SNNs is challenging due to the non-differentiability of the spike activity. The backpropagation through time with surrogate gradient method [25] has bridged neuromorphic computing with the deep learning community, enabling the training of deeper SNNs. Despite this advancement, the performance of SNNs still lags behind that of ANN methods.

The Visual Transformer (ViT) and its variant architectures have garnered increasing interest as potential replacements for convolution networks in various computer vision tasks in recent years. Traditional convolution-only models struggle to capture temporal correlation and efficiently represent global spatial dependencies due to their inherent locality [4, 11, 15, 29]. The integration of ViT, particularly with spatiotemporal attention, has also shown promising results in event-based vision tasks, such as monocular depth estimation [42] and action recognition [1, 8]. Combining SNNs with the ViT architecture for event cameras appears to be a natural choice, promising both performance and energy efficiency. Moreover, the self-attention mechanism in transformers also shares a biological background with SNNs [34, 39, 40, 43]. While early work proposing the Spikeformer architectures primarily validates their efficacy on event-based classification tasks [39, 43], their application in event-based regression tasks remains limited [47].

Solutions for event-based optical flow are primarily dominated by correlation-based methods [10, 19, 30], which require substantial computation and memory resources to compute the pixel-wise correlation volume. Integrating transformers into optical flow models has shown superior performance compared to non-transformer models in conventional computer vision, particularly excelling in scenarios involving large displacements due to their ability to capture global dependencies effectively [15, 23, 36]. While some works have utilized transformer architectures for event camera optical flow estimation tasks [18, 32], no one has proposed a pure SNN architecture, specifically utilizing spikeformer, for event-based optical flow estimation.

In this work, we introduce SDformerFlow, an SNN employing spiking spatiotemporal Swin transformers. Additionally, for better comparison, we propose STTFlowNet, an ANN counterpart to our SNN model. We conduct end-to-end training using supervised learning on the DSEC-flow Dataset. Our work marks the first instance of utilizing spikeformer for optical flow estimation, demonstrating comparable performance to state-of-the-art SNN optical flow estima-

tion while significantly reducing energy consumption compared to the baseline model. Our contributions are threefold: Firstly, we introduce STTFlowNet, a Swin transformer-based model for event-based optical flow estimation, equipped with spatiotemporal self-attention to capture dependencies in both the time and space domains. Secondly, we present the spiking version of our architecture, SDformerFlow, marking the first known utilization of spikeformer for event-based optical flow estimation. Lastly, we conduct extensive experiments on datasets and compare them with baseline models, uncovering the potential of combining transformers with SNNs for regression tasks.

## 2 Related work

### 2.1 Learning-based methods for event-based optical flow estimation

Drawing inspiration from frame-based optical flow techniques, the estimation of event-based optical flow using deep learning has achieved state-of-the-art performance compared to model-based methods [10, 26, 28]. Early works predominantly employed a U-Net architecture [3, 12, 17, 46] to predict sparse flow and evaluated it using masks due to limited accuracy where no events are present. Inspired by RAFT flow [30], Gehrig et al. [10] proposed E-RAFT and contributed the Dense Stereo Event Camera (DSEC) Optical Flow Benchmark [9]. Since then, methods based on recurrent neural networks with correlation features and iterative refinement strategies have yielded state-of-the-art performance [3, 10, 18].

Recent studies have shifted their focus towards enhancing the temporal continuity of optical flow estimation, aiming to fully leverage the low latency characteristics of event cameras [19, 27, 35], or integrating richer simulated training datasets [18, 37] to improve accuracy. However, these recurrent refinement methods implicate calculating computationally expensive cost columns and an iterative update scheme that brings latency to the inference phase.

Another line of work based on SNNs emerges as a computationally efficient solution for event camera optical flow estimation. Early works trained SNNs using self-supervised learning, yielding sparse flow estimation [12]. More recent efforts involve training SNNs using supervised learning with U-Net architectures and trained with surrogate gradient on the DSEC dataset, resulting in dense flow estimation [2, 16]. To incorporate longer temporal correlations into the SNN model, some works utilize adaptive neural dynamics in comparison with event inputs containing richer temporal information [16], while others introduce external recurrence [27]. In [2], the authors employed 3D convolutions with stateless spiking neurons, neglecting the intrinsic temporal dynamics of the neurons. However, the performance of SNNs still falls behind that of ANNs. While some ANN methods incorporate transformer architectures in some of their stages [18, 32] and show performance improvements, to the best of our knowledge, this is the first time that SNNs are combined with a transformer architecture for optical flow estimation.

## 2.2 Spikeformer

Recently, the combination of SNNs and transformer architectures has garnered increasing interest for other tasks in the neuromorphic community [38, 40, 43]. Zhou et al. [43] initially proposed spiking self-attention, which eliminates the softmax function as the spike-formed query and key naturally maintains non-negativity. Building upon this, Yao et al. [39] introduced a fully spike-driven transformer with spike-driven self-attention, leveraging only mask and addition operations to facilitate hardware implementation. Recently, Yao et al. [38] extended their previous work [39] into a meta-architecture that achieves state-of-the-art results in SNN classification, detection, and segmentation tasks. While most spikeformers only apply spatial-wise attention in a single time step [43], some works also incorporate spatiotemporal attention [33, 47]. However, none of the previous works have utilized the Swin variant of the Spikeformer, nor for optical flow estimation.

## 3 Method

### 3.1 Event Input Representation

We divide the event stream into non-overlapping chunks. Each chunk, comprising  $N$  events within a fixed time window, is represented as  $E = \{(x_i, y_i, t_i, p_i)\}_{i \in [N]}$ . We preprocess each event chunk into an event discretized volume representation  $V$  using a set of bins  $B$ , following the methodology introduced in [46].

$$V(x, y, t) = \sum_i p_i \kappa(x - x_i) \kappa(y - y_i) \kappa(t - t_i) \quad (1)$$

Timestamps are normalized and scaled to the range  $[0, B - 1]$ ,  $t_i = (B - 1)(t_i - t_0)/(t_N - t_1)$ ; and  $\kappa(a) = \max(0, 1 - |a|)$  is a bilinear sampling kernel. To enable the neural network to learn large temporal correlations, we encode spatiotemporal information into channels. For the ANN model, we take the previous and current chunks of event voxels, dividing the total temporal channels into  $n$  blocks. Each event input block comprises  $2B/n \times H \times W$  bins. In our case,  $n = 2$ .

For the SNN model, to mitigate the computational burden associated with large time steps, we use only one event voxel chunk. Similarly, we partition the temporal channel, containing  $B$  bins, into  $n$  blocks along with their corresponding polarities  $p$ . This yields an event representation of size  $T \times 2n \times H \times W$ , with  $T = B/n$  time steps. In our implementation, we set  $B = 10$  and  $n = 2$ . This representation aligns with the spike representation outlined in [16, 17]. Each event chunk comprises  $C = 4$  channels and  $T = B/2$  time steps, as illustrated in Fig. 1.

### 3.2 Spiking Neuron

We utilize the Leaky Integrate-and-Fire (LIF) neuron model for all layers in our models. LIF is widely adopted in the literature due to its simplicity of implementation and low computational cost. For our implementation, we employ the Spikejelly library [6] and set  $V_{th} = 0.1$  and  $\tau_m = 2$ .

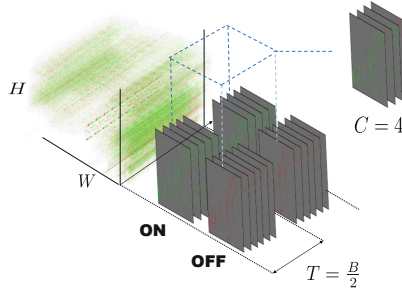


Fig. 1. Event input representation.

### 3.3 Network Architecture

The network architecture pipeline of our proposed methods STTFlowNet and SDformerFlow (Fig. 2) is similar. We adopt an encoder-decoder architecture, widely utilized in event-based optical flow literature [2, 12, 44, 46]. For STTFlowNet, the architecture of the Swin Transformer encoder resembles that of [22]. Each Swin block contains a Multi-Head Self-Attention (MSA) module, followed by a Feed-Forward Network (FFN) module consisting of two MLP layers. Layer Normalization (LN) is applied after each module, with residual connections incorporated. In the MSA module, unlike previous implementations [22], we utilize scaled cosine attention and logarithmic continuous relative position bias (CPB) from Swin Transformer V2 [20] to enhance the model’s scaling capability. In the following sections, we focus on detailing the architecture of SDformerFlow. Further details are provided in the supplementary document.

For SDformerFlow, the primary architecture comprises three components: a) a Spike Feature Generator (SFG) embedding module, b) a Spatiotemporal Swin Spikeformer (STSF) encoder, and c) spike decoders and flow prediction. The event stream initially enters the SFG module, which outputs spatiotemporal embeddings for the STSF encoders. The STSF encoders then generate spatiotemporal features hierarchically. Subsequently, the output from each encoder is concatenated to the decoder at the same scale to predict the flow map. Two additional residual blocks exist between the encoder and decoder modules.

We propose two variants of our fully spiking model: SpikeformerFlow and SDformerFlow. The key distinction lies in the residual connection. In SpikeformerFlow, all residual shortcuts utilize spike-element-wise shortcuts (SEW) [7]. Conversely, in SDformerFlow, we employ membrane-potential shortcut (MS) [14]. Figure 3(b) illustrates the main differences among the vanilla shortcut, SEW shortcuts, and MS shortcuts. The vanilla shortcut adds spikes into the memory potential values, which cannot achieve identity mapping and have degradation problem [14]. In SEW shortcuts, the residuals are applied after the spikes, which results in integrals. Meanwhile, in MS shortcuts, residuals are applied before the spikes to preserve the spike-driven property.

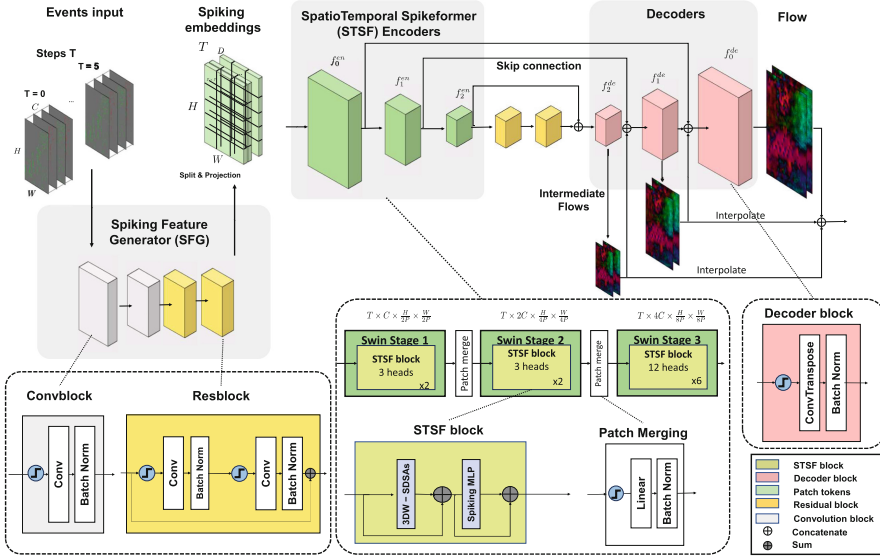


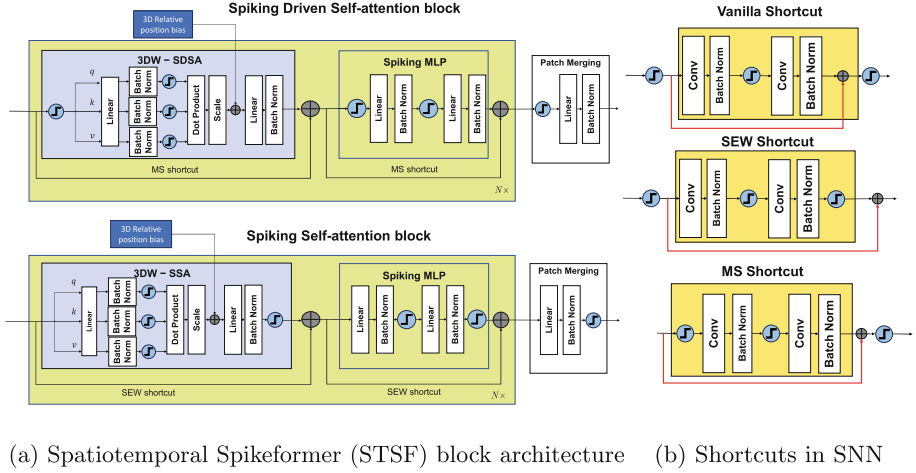
Fig. 2. SDformerFlowNet architecture.

**Spiking Feature Generator Embedding (SFG)** The SFG module comprises two stages: generating spatiotemporal features and projecting them into token embeddings for the STSF encoder module. In the first stage, we process the event input through a spiking convolutional module followed by two residual blocks to downsample the resolution by half. This projection results in a feature map of shape  $T \times C \times H/2 \times W/2$ . In the second stage, we split the feature map into spatial patches of size  $P \times P$ , maintaining the time steps as the temporal dimension. This operation creates spatiotemporal tokens of size  $1 \times P \times P$ , projecting the spatial-temporal features into spike embeddings of shape  $T \times C \times H/(2P) \times W/(2P)$ . For STTFlowNet, both the former and latter chunks are fed into a shared *Resblock* module while retaining the spatial dimension.

**Spatiotemporal Swin Spikeformer (STSF) Encoder** The STSF module draws inspiration from the Video Swin Transformer [22] and Spikeformer [43]. Its detailed architecture is illustrated in Fig. 2.

We adopt three stages of Swin Transformer, with each stage comprising 2-2-6 numbers of STSF blocks successively, followed by a patch merging layer to reduce the dimension by half.

Each STSF block comprises a spiking multi-head Spike-driven Self-Attention (SDSA) block with a 3D shifted window (3DW), followed by a spiking MLP block (see Fig. 3(a) upper). Each spatiotemporal token of shape  $T \times H \times W$  is partitioned into non-overlapping 3D windows of size  $T_w \times H_w \times W_w$ . We employ a window size of  $2 \times 9 \times 9$  for cropped resolution and  $2 \times 15 \times 15$  when fine-tuning



**Fig. 3.** In (a): The top diagram depicts the Spike-driven Self-Attention (SDSA) block utilized in SDFormerFlow, while the bottom one illustrates the Spiking Self-Attention (SSA) block with Spike-Element-Wise (SEW) shortcuts used in Spikeformerflow and in other architectures [43]. Fig.(b) shows the comparison of different residual shortcuts in SNN.

the model on a full resolution of  $480 \times 640$ . The SDSA is performed within the window. We utilize different numbers of attention heads (3, 6, 12) for the STSF blocks in different stages. The details of our SDSA and spiking patch merge modules are explained as follows:

*Spike-Driven Self-Attention (SDSA)* In our SDSA block, the Query, Key, and Value tensors, denoted as  $Q_s, K_s, V_s$ , are spiking tensors. We use dot product attention, and since the attention maps are naturally non-negative, softmax is unnecessary [43]. We additionally apply a scale factor  $s$  for normalization to prevent gradient vanishing. The single-head SDSA can be formalized as follows:

$$SDSA(Q_s, K_s, V_s) = BN(Linear(SN(Q_s K_s^T V_s * s))) \quad (2)$$

*Spiking Patch Merge* The patch merge layer comprises a Linear layer followed by a batch normalization layer to downsample the feature map in the spatial domain.

**Spike Decoder Block** The decoder consists of three Transposed convolutional layers, each increasing the spatial resolution by a factor of two. A skip connection from each STSF encoder is concatenated to the prediction output from the corresponding decoder of the same scale. Flow prediction is generated at each scale and concatenated to the decoders. Loss is applied to the flow prediction upsampled to the full resolution.

### 3.4 Loss Function

We train our model with supervised learning using the mean absolute error between the estimated optical flow  $\mathbf{u}_i^{pred} = (u_i^{pred}, v_i^{pred})$  and the ground-truth flow  $\mathbf{u}_i^{gt} = (u_i^{gt}, v_i^{gt})$ . Our loss function can be formulated as:

$$L = \frac{1}{n} \sum_{i=1}^n |\mathbf{u}_i^{pred} - \mathbf{u}_i^{gt}| \quad (3)$$

where  $n$  is the number of valid ground truth pixels. For SNN, we employ surrogate gradient (SG) [25] with backpropagation through time (BPTT) to train the network. We use the inverse tangent as the surrogate function with a width of 2.

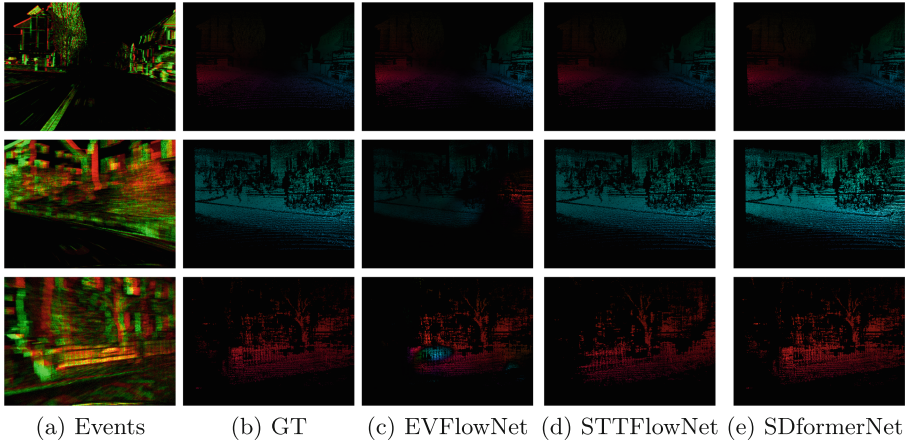
## 4 Experiments

### 4.1 Dataset and training details

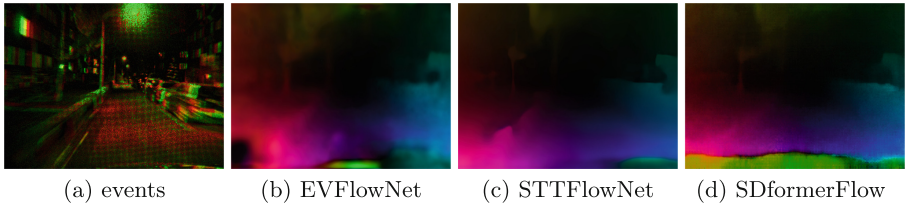
We utilize the DSEC dataset [9] for both training and evaluation purposes. The DSEC dataset is a comprehensive outdoor stereo event camera dataset featuring a resolution of  $640 \times 480$ . Ground-truth optical flow annotations are provided at a rate of 10Hz for some of the sequences. To address the lack of ground truth in the test set, we adopt a similar data split strategy to [2], dividing the training sequences into training and validation sets. Notably, we exclusively utilize rectified event data from the left camera. During training and validation, we perform data augmentation techniques, including random horizontal and vertical flips, as well as random crops on a  $288 \times 384$  resolution. We train the models on three NVIDIA GeForce RTX 2080 Ti GPUs and employ the AdamW optimizer for a total of 80 epochs, ensuring convergence. The initial learning rate is set to 0.001 with a weight decay of 0.01. Additionally, we implement a multistep scheduler that halves the learning rate every 10 epochs. To mitigate performance degradation when scaling up to the full resolution, we conduct fine-tuning on the full-resolution data for an additional 30 epochs before testing. Given the constraints of GPU memory, training at full resolution necessitates a reduced batch size (1 or 2). During evaluation, we disable the tracking of running states for batch normalization layers to optimize memory usage.

Additionally, we trained our models on the MVSEC dataset [45]. Since the MVSEC dataset and DSEC dataset share different spatial resolutions and ground truth rates. We trained our model using MDR training dataset [24] with cropped resolution  $256 \times 256$  and reported our evaluation results for the sparse optical flow to compare with other models.

## 4.2 Results



**Fig. 4.** Qualitative results for optical flow are evaluated on the DSEC validation dataset. The first column displays the event input, while the second column depicts the ground truth dense optical flow from our split validation dataset. During evaluation, we mask the estimated flow where ground truth flows are available. (Best viewed in color).



**Fig. 5.** Qualitative results for optical flow are evaluated on the official DSEC test dataset. The first column presents the event input, while the other columns show the corresponding estimated optical flow for the baseline method EVFlow and our methods STTFlowNet and SDformerFlow. (Best viewed in color).

The quantitative results for our models, STTFlowNet and SDformerFlow, evaluated on the DSEC benchmark, are presented in Table 1. Additionally, Figures 4 and 5 illustrate the qualitative results obtained from the validation and test dataset, respectively.

Figure 4 shows both our STTFlowNet and SDformerFlow models, trained with cropped resolution on our split training dataset and tested on the validation



**Table 1.** Quantitative results for optical flow estimation of the DSEC optical flow benchmarks for all the test sequences. The first column shows the methods, A stands for ANN, S stands for SNN, while M stands for model-based method.

Training		EPE	Outlier %	AAE
A	E-RAFT [10]	0.779	2.684	2.838
	EV-FlowNet_retrained [10]	2.32	18.60	-
	IDNet [35]	0.719	2.036	2.723
	TMA [19]	0.743	2.301	2.684
	E-Flowformer [18]	0.759	2.446	2.676
	TamingCM[26]	2.33	17.771	10.56
	STTFlowNet-en3 (Ours)	0.997	4.588	3.235
S	SNN_3DNet[2]	1.707	10.308	6.338
	SDFormerFlow-en3 (Ours)	2.142	14.021	5.941
M	MultiCM [28]	3.472	30.855	13.983

**Table 2.** Quantitative results for optical flow evaluated for MVSEC dataset. D shows the training dataset: MVSEC dataset, FPV or MDR dataset. We highlight the best-performing results and underline the best results for the SNN model in each tested sequence.

Training	dt = 1 frame	D	outdoor_day1	indoor_flying1	indoor_flying2	indoor_flying3	Avg					
			AEE %	Outlier	AEE %	Outlier	AEE %	Outlier	AEE %	Outlier	AEE %	Outlier
A	EV-FlowNet [44]	M	0.49	0.20	1.03	2.20	1.72	15.10	1.53	11.90	1.19	7.35
	EV-FlowNet2 [46]	M	<b>0.32</b>	0.00	0.58	0.00	1.02	4.00	0.87	3.00	0.69	1.75
	GRU-EV-FlowNet [12]	FPV	0.47	0.25	0.60	0.51	1.17	8.06	0.93	5.64	0.79	3.62
	STE-FlowNet [3]	M	0.42	0	0.57	0.1	0.79	1.6	1.72	1.3	0.62	0.75
	ET-FlowNet [32]	FPV	0.39	0.12	0.57	0.53	1.2	8.48	0.95	5.73	0.78	3.72
	ADM-Flow [24]	MDR	0.41	0.00	<b>0.52</b>	0.14	<b>0.68</b>	1.18	<b>0.52</b>	0.04	<b>0.53</b>	0.34
	STT-FlowNet (ours)	MDR	0.66	0.29	0.57	0.33	0.88	4.47	0.73	1.58	0.71	1.67
S	Spike-FlowNet [17]	M	0.49		0.84		1.28		1.11		0.93	
	XLIF-EV-FlowNet [12]	FPV	0.45	0.16	0.73	0.92	1.45	12.18	1.17	8.35	0.95	5.40
	Adaptive-SpikeNet [16]	FPV	<u>0.44</u>		0.79		1.37		1.11		0.93	
	SNN3DNet [2]	M	0.85		<u>0.58</u>		<u>0.72</u>		<u>0.67</u>		<u>0.71</u>	
	SDFormerFlow (Ours)	MDR	0.69	0.21	0.61	0.60	0.83	3.41	0.76	1.45	0.72	1.42

dataset. We use bicubic interpolation to remap the relative positional bias for full resolution, as described in [21]. Notably, when the vehicle moves forward in steady motion, all models achieve accurate flow estimation. However, in scenarios involving sharp turns or large, abrupt motions (third row in the figure), the baseline EVFlowNet struggles to estimate the correct direction. In contrast, both our STTFlowNet and our fully spiking model effectively handle such scenarios, thanks to their utilization of spatiotemporal attention mechanisms.

Figure 5 showcases the improved estimation performance of our models on the DSEC optical flow benchmark<sup>1</sup> test set compared to the baseline. Notably, SDformerFlow achieves superior estimation, although it encounters challenges in areas where the sensor hits the car hood for which ground truth data is unavailable. This limitation could be attributed to the additional convolutional layers added in the early stages to downsample and reduce memory consumption.

Regarding the quantitative evaluation presented in Table 1, our ANN model outperforms the baseline model [46] and other self-supervised trained models [26]. However, it still trails behind correlation-volume-based models [10, 18]. The only SNN model included in the benchmark [2] uses stateless neurons and is trained at full resolution, whereas most other SNN approaches are trained and validated on cropped resolution [16, 27], with limited representation in the benchmark. Notably, our fully spiking model, SDformerFlow, exhibits superior performance compared to the ANN baseline [46].

The quantitative evaluation tested on MVSEC dataset is presented in Table 2. Both our ANN and SNN models yield competitive results. Our ANN model performs better than another transformer-based U-Net architecture [32]. Our SDformerFlow ranked second for the average AEE for all the sequences among all the SNN methods. However, the best performing model [2] reports their results for the indoor sequences separately trained on the subsets of the same dataset, which may have overfitted to the test dataset.

### 4.3 Ablation studies

**Table 3.** Ablation study for STTFlowNet. Column I stands for the event input type. For the variant of STTFlowNet, en means number of encoders, b stands for number of input blocks, p means spatial patch size, and w stands for swin spatial window size. Best-performing results are highlighted.

Model	EPE	Outlier %	AAE	I	Training res.	Param. (M)
EVFlowNet_retrained	1.63	10.01	5.84	count	288,384	14.14
EVFlowNet_retrained	1.57	9.918	6.09	voxel	288,384	14.14
STTFlowNet-en3-b2-p4-w5	1.67	12.61	8.22	count	240,320	20.30
STTFlowNet-en3-b2-p2-w10	1.34	8.29	5.98	count	240,320	20.30
STTFlowNet-en3-b4-p4-w10	1.37	8.21	6.77	count	240,320	20.29
STTFlowNet-en3-b4-p2-w10	1.43	9.44	5.54	count	240,320	20.29
STTFlowNet-en3-b4-p2-w10	1.05	4.97	5.34	voxel	240,320	20.29
STTFlowNet-en3-b2-p2-w10	0.94	3.97	4.78	voxel	240,320	20.30
STTFlowNet-en3-b2-p4-w10	0.83	2.61	4.36	voxel	480,640	20.29
STTFlowNet-en4-b2-p4-w10	<b>0.81</b>	<b>2.50</b>	<b>4.33</b>	voxel	480,640	57.51

<sup>1</sup> Full benchmark statistics are available at <https://dsec.ifi.uzh.ch/uzh/dsec-flow-optical-flow-benchmark/>

**Table 4.** Ablation study for SDformerFlow. For the SNN model variants, en denotes number of encoders, s stands for number of steps, and c stands for number of channels. Best performing results are highlighted in bold.

Model	EPE		Outlier %		AAE		I	Training res.	Param. (M)
	C	F	C	F	C	F			
test res: cropped (C) or full (F)	C	F	C	F	C	F			
LIF-EV-FlowNet-en4-s5	3.08	3.47	19.67	23.70	17.90	14.41	voxel10	288,384	14.13
SpikeformerFlowNet-SEW-en3-s8-c4	1.60	3.21	11.90	32.30	12.51	14.77	voxel15*	240,320	19.80
SpikeformerFlowNet-SEW-en3-s4-c8	1.76	3.54	13.43	41.18	14.01	27.81	voxel15*	240,320	19.81
SpikeformerFlowNet-SEW-en3-s5-c4	1.51	2.52	9.85	22.75	10.68	11.10	voxel10	288,384	19.83
SpikeformerFlowNet-MS-en3-s5-c4	1.28	2.01	6.91	15.55	9.01	8.99	voxel10	288,384	19.83
SpikeformerFlowNet-MS-en4-s5-c4	<b>1.25</b>	<b>1.98</b>	<b>6.69</b>	<b>15.06</b>	<b>8.48</b>	<b>8.81</b>	voxel10	288,384	56.48
SpikeCAformerFlow-MS-en4-s5-c4	1.66	2.97	10.65	27.87	12.05	22.55	voxel10	288,384	15.73

\*The SEW variant with input voxel size of 15 was trained with a resolution of  $240 \times 320$  due to GPU memory limitations. The rest of the Spikeformer models were trained at  $288 \times 384$  resolution.

The ablation study was conducted on the validation DSEC set. For the ANN models, we retrained EVFlowNet [46] on the DSEC training set as our base model for 60 epochs while randomly cropping to size  $288 \times 384$ . Our ANN model shares the same U-Net architecture with EVFlowNet, with the key difference being the use of spatiotemporal swin transformer encoders instead of convolutional layers. Our models were trained at either half or full resolution of  $480 \times 640$ , and validated in full resolution. We analyzed the effects of: a) the input representation: event voxel or count; b) the number of temporal partitioning blocks: 2 (b2) or 4 (b4); c) the spatial patch size:  $p = 2$  or  $p = 4$ , and swin spatial window size  $w$ ; and d) the training resolution.

Results are summarized in Table. 3. Using the event voxel representation retained more temporal information and notably improved results. Introducing swin transformer layers instead of convolutions also led to significant performance gains. For the variants of STTFlowNet, the window size influenced the range of the area to pay attention to, with smaller window sizes making it difficult for the network to learn larger displacements. Adjusting the patch size between 2 and 4 according to the window size and resolution was found to be effective. Partitioning the temporal domain into 2 chunks yielded better results than 4, potentially due to the total number of channels. Further improvements may be achieved by incorporating a local-global chunking approach as described in [42]. To maintain equivalence between our ANN and SNN models, we utilized local temporal blocks exclusively. Given the performance degradation experienced by the swin transformer at higher resolutions, we opted to train the model directly at full resolution using a patch size of 4. This approach ensured that the resolution within the swin encoders remained consistent with training the model at half resolution with a patch size of 2. Notably, this strategy resulted in remarkable improvements in performance.

For our SNN model, we trained the fully spiking version of EVFlowNet with LIF neurons using the same input representations as our base model for com-

**Table 5.** Energy consumption for ANN and SNN models

Model	EPE	Type	Param (M)	FLOPS(G)	Avg. spiking rate	Power(mJ)
EVFlowNet retrained	1.57	ANN	14.14	22.38	-	102.95
LIF-EVFlowNet	3.08	SNN	14.13	22.38	0.29	29.21
STTFlowNet-en3	0.72	ANN	20.30	86.88	-	399.65
SDFlowNet-en3	1.28	SNN	19.83	34.80	0.27	37.64
SDFlowNet-en4	1.25	SNN	56.48	39.10	0.27	41.08

parison. We studied: a) the number of time steps/channels; b) shortcut variants: SEW or MS shortcuts; and c) the number of encoders.

Results are presented in Table. 4. The spikeformer encoders significantly improved performance compared to the baseline model, albeit with reduced robustness when directly tested on scaled-up resolutions. Increasing the number of time steps helped capture temporal information at the expense of increased memory consumption. Opting for 5 time steps and 4 channels struck a balance between performance and memory consumption. The MS shortcut variant notably improved results compared to SEW shortcut. One possible explanation is that the MS shortcut provides an information flow path between the states of the neurons before the spike function and is not regulated by their firing status. Increasing the number of encoders from three to four further enhanced performance at the cost of increased parameters. Finally, incorporating convolution-based modules as CAformer in [38,41] in the first two swin encoders yielded a lightweight model but with slightly reduced performance.

#### 4.4 Energy consumption analysis

In our energy consumption analysis, we follow established methodologies from prior research [16,27,38]. For ANN models, we estimate energy consumption based on the number of floating-point operations (FLOPS) required, as all operations in ANN layers are multiply-accumulate (MAC) operations. Therefore, the energy consumption for ANN models is calculated as  $FLOPS \times E_{MAC}$ . Conversely, SNN models convert multiplication operations into addition operations due to their binary nature. Thus, for SNN models, we estimate energy consumption by multiplying the FLOPS with the spiking rate  $R_s$  and the number of time steps  $T$ , resulting in  $FLOPS \times R_s \times T \times E_{AC}$ . Here,  $E_{MAC}$  represents the energy required for MAC operations, and  $E_{AC}$  represents the energy required for addition operations. For 32-bit floating-point computation, these energy values are typically  $E_{MAC} = 4.6pJ$  and  $E_{AC} = 0.9pJ$ , respectively, based on a 45 nm technology [13]. We estimate the average spiking rates among all time steps for each layer to calculate energy consumption, ignoring the negligible contribution of batch normalization layers (around 0.01%). The energy consumption for each model during the inference phase, with an image input size of  $288 \times 384$ , is presented in Table 5. Our results demonstrate that the energy consumption of our

SNN model is nearly one-tenth that of its ANN counterpart and one-third that of the baseline EVFlowNet model.

## 5 Conclusions and future work

We introduced STTFlowNet and SDformerFlow, two novel architectures for event-based optical flow estimation that leverage spatiotemporal swin transformer encoders in ANN and SNN frameworks, respectively. Our work marks the first application of a spikeformer for event-based optical flow estimation. Despite not using correlation volumes and facing scalability challenges inherent to transformer architectures, our results highlight the potential in the use of spikeformers in regression tasks. Our ablation studies shed light on the importance of key components such as input representation, temporal partitioning, and spatial window size, providing insights for future research directions. Our SNN version is the first fully spikeformer implementation and is comparable to the other SNN implementations reported in the benchmark. Notably, our SNN model achieved remarkable energy savings compared to its ANN counterpart and also outperformed the baseline EVFlowNet model. We believe by introducing spatiotemporal attention, we strengthen our model's capability to map global context for the spatial feature maps while capturing spatiotemporal correlations, which improves the performance of our model compared to other CNN-based methods.

However, one limitation of our work is that, by feeding the entire chunk of data into the spatiotemporal attention modules, we were not able to fully exploit the asynchronous ability of the event camera and SNNs. This can be improved by introducing temporal delay, as proposed in [33], in future work. Secondly, transformer-based models suffer from constrained scalability across different resolutions. Recent work proposes methods to address this issue by incorporating multi-resolution training [31] or dynamic resolution adjustment modules [5]. Thirdly, our model is based on the encoder-decoder architecture to prevent calculating the correlation features and iterative process since it obeys the motivation to use SNN as a computational and energy-efficient solution, thus the performance still falls behind some state-of-the-art methods. For improvement, we propose to train with more diverse datasets and exploit learned neural dynamics parameters. Finally, much work remains to be done related to hardware implementation to fully exploit the advantage of energy efficiency of SNNs.

In conclusion, our work highlights the efficacy of integrating transformer architectures with spiking neural networks for efficient and robust optical flow estimation, paving the way for advancements in neuromorphic vision systems.

**Acknowledgments.** This work received support from projects EBCON (PID2020-119244GB-I00) and RAADICAL (PLEC2021-007817) funded by MCIN/ AEI/ 10.13039/ 501100011033 and the "European Union NextGenerationEU/PRTR"; the Consolidated Research Group RAIG (2021 SGR 00510) of the Departament de Recerca i Universitats de la Generalitat de Catalunya; and by an FI AGAUR PhD grant to Yi Tian.

## References

1. de Blegiers, T., Dave, I.R., et al.: Eventtransact: A video transformer-based framework for event-camera based action recognition. In: IEEE/RSJ Int. Conf. Intell. Robots Syst. pp. 1261–1267 (2023)
2. Cuadrado, J., Rançon, U., et al.: Optical flow estimation from event-based cameras and spiking neural networks. *Front. Neurosci.* **17**, 1160034 (2023)
3. Ding, Z., Zhao, R., et al.: Spatio-temporal recurrent networks for event-based optical flow estimation. In: AAAI Conf. Artif. Intell. vol. 36, pp. 525–533 (2021)
4. Dosovitskiy, A., Beyler, L., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *Int. Conf. Learn. Represent.* (2020)
5. Fan, Q., You, Q., et al.: Vitar: Vision transformer with any resolution. *arXiv preprint [arXiv:2403.18361](https://arxiv.org/abs/2403.18361)* (2024)
6. Fang, W., Chen, Y., et al.: Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence. *Sci. Adv.* **9**(40), eadi1480 (2023)
7. Fang, W., Yu, Z., et al.: Deep residual learning in spiking neural networks. In: *Conf. Neural Inf. Process. Syst.* vol. 34, pp. 21056–21069 (2021)
8. Gao, Y., Lu, J., et al.: Action recognition and benchmark using event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(12), 14081–14097 (2023)
9. Gehrig, M., Aarents, W., et al.: DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics Autom. Lett.* **6**(3), 4947–4954 (2021)
10. Gehrig, M., Millhauser, M., et al.: E-RAFT: Dense optical flow from event cameras. In: *Int. Conf. 3D Vis.* pp. 197–206 (2021)
11. Guizilini, V., Ambrus, R., et al.: Multi-frame self-supervised depth with transformers. In: *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* pp. 160–170 (2022)
12. Hagenars, J., Paredes-Vallés, F., de Croon, G.: Self-supervised learning of event-based optical flow with spiking neural networks. In: *Conf. Neural Inf. Process. Syst.* vol. 34 (2021)
13. Horowitz, M.: 1.1 computing’s energy problem (and what we can do about it). In: *IEEE Int. Solid-State Circuits Conf.* pp. 10–14 (2014)
14. Hu, Y., Deng, L., et al.: Advancing spiking neural networks toward deep residual learning. *IEEE Trans. Neural Networks Learn. Syst.* (2024), early access
15. Huang, Z., Shi, X., et al.: Flowformer: A transformer architecture for optical flow. In: *Eur. Conf. Comput. Vis.* pp. 668–685 (2022)
16. Kosta, A.K., Roy, K.: Adaptive-spikenet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics. In: *IEEE Int. Conf. Robotics Autom.* pp. 6021–6027 (2023)
17. Lee, C., Kosta, A., et al.: Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In: *Eur. Conf. Comput. Vis.* pp. 366–382 (2020)
18. Li, Y., Huang, Z., et al.: Blinkflow: A dataset to push the limits of event-based optical flow estimation. In: *IEEE/RSJ Int. Conf. Intell. Robots Syst.* pp. 3881–3888 (2023)
19. Liu, H., Chen, G., et al.: TMA: Temporal motion aggregation for event-based optical flow. In: *IEEE Int. Conf. Comput. Vis.* pp. 9651–9660 (2023)
20. Liu, Z., Hu, H., et al.: Swin transformer v2: Scaling up capacity and resolution. In: *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* pp. 12009–12019 (2022)
21. Liu, Z., Lin, Y., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: *IEEE Int. Conf. Comput. Vis.* pp. 10012–10022 (2021)

22. Liu, Z., Ning, J., et al.: Video swin transformer. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 3202–3211 (2022)
23. Lu, Y., Wang, Q., et al.: Transflow: Transformer as flow learner. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 18063–18073 (2023)
24. Luo, X., Luo, K., et al.: Learning optical flow from event camera with rendered dataset. arXiv preprint [arXiv:2303.11011](https://arxiv.org/abs/2303.11011) (2023)
25. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* **36**(6), 51–63 (2019)
26. Paredes-Vallés, F., Scheper, K.Y.W., et al.: Taming contrast maximization for learning sequential, low-latency, event-based optical flow. In: IEEE Int. Conf. Comput. Vis. pp. 9695–9705 (2023)
27. Ponghiran, W., Liyanagedera, C.M., Roy, K.: Event-based temporally dense optical flow estimation with sequential learning. In: IEEE Int. Conf. Comput. Vis. pp. 9827–9836 (2023)
28. Shiba, S., Aoki, Y., Gallego, G.: Secrets of event-based optical flow. In: Eur. Conf. Comput. Vis. pp. 628–645 (2022)
29. Sui, X., Li, S., et al.: CRAFT: Cross-attentional flow transformer for robust optical flow. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 17581–1790 (2022)
30. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Eur. Conf. Comput. Vis. pp. 402–419 (2020)
31. Tian, R., Wu, Z., et al.: Resformer: Scaling vits with multi-resolution training. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 22721–22731 (2023)
32. Tian, Y., Andrade-Cetto, J.: Event transformer flownet for optical flow estimation. In: British Mach. Vis. Conf. (2022)
33. Wang, Y., Shi, K., et al.: Spatial-temporal self-attention for asynchronous spiking neural networks. In: Int. Joint Conf. Artif. Intell. pp. 3085–3093 (2023)
34. Wang, Z., Fang, Y., et al.: Masked spiking transformer. In: IEEE Int. Conf. Comput. Vis. pp. 1761–1771 (2023)
35. Wu, Y., Paredes-Vallés, F., de Croon, G.C.H.E.: Rethinking event-based optical flow: Iterative deblurring as an alternative to correlation volumes. arXiv preprint [arXiv:2211.13726](https://arxiv.org/abs/2211.13726) (2023)
36. Xu, H., Zhang, J., et al.: Gmflow: Learning optical flow via global matching. In: IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 8121–8130 (2022)
37. Yang, Y., Pan, L., Liu, L.: Event camera data pre-training. arXiv preprint [arXiv:2301.01928](https://arxiv.org/abs/2301.01928) (2023)
38. Yao, M., Hu, J., et al.: Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips. In: Int. Conf. Learn. Represent. (2024)
39. Yao, M., Hu, J., et al.: Spike-driven transformer. In: Conf. Neural Inf. Process. Syst. (2023)
40. Yao, M., Zhao, G., et al.: Attention spiking neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(8), 9393–9410 (2023)
41. Yu, W., Si, C., et al.: Metaformer baselines for vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(2), 896–912 (2024)
42. Zhang, J., Tang, L., et al.: Spike transformer: Monocular depth estimation for spiking camera. In: Eur. Conf. Comput. Vis. pp. 34–52 (2022)
43. Zhou, Z., Zhu, Y., et al.: Spikformer: When spiking neural network meets transformer. In: Int. Conf. Learn. Represent. (2023)
44. Zhu, A., Yuan, L., et al.: EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In: Robotics Sci. Syst. Conf. (2018)

45. Zhu, A.Z., Thakur, D., et al.: The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robotics and Automation Letters* **3**(3), 2032–2039 (2018)
46. Zhu, A.Z., Yuan, L., et al.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* pp. 989–997 (2019)
47. Zou, S., Mu, Y., et al.: Event-based human pose tracking by spiking spatiotemporal transformer. arXiv preprint [arXiv:2303.09681](https://arxiv.org/abs/2303.09681) (2023)



# Author Index

## A

Aakur, Sathyanarayanan N. 293  
Ahmad, Niaz 145  
Al-Saad, Mohammad 407  
Andrade-Cetto, Juan 475  
Aysa, Alimjan 129

## B

Barua, Hrishav Bakul 343  
Bhandarkar, Suchendra M. 407  
Bosetti, Massimo 327  
Burns, Matthew 212

## C

Chen, Weihao 392  
Chen, Zhongqi 245  
Cheng, Jiale 278  
Choi, Chanyeok 145

## D

Dass, Sharana Dharshikgan Suresh 343  
Du, Guocai 129

## F

Fernando, Basura 162  
Fink, Gernot A. 17, 33

## G

Gajbhiye, Gaurav 375  
Goyal, Tanish 96  
Grzeszick, René 17  
Guo, Yu 458

## H

Hallyburton, Tim 17  
Haque, Syed Tousiful 425  
Huang, Qian 245

## J

Ji, Yantao 458  
Jiang, Peilin 458  
Jiang, Yaru 360  
Jutla, Charanjit 49, 65

## K

Kampel, Martin 178, 194  
Kaushik, Arjun Ramesh 49  
Kawamura, Ryosuke 375  
Khan, Jawad 145  
Kiftiyani, Usfita 310  
Kimura, Keigo 80  
Kolekar, Maheshkumar 1  
Krishnasamy, Ganesh 343  
Kudo, Mineichi 80  
Kumar, Sanjeev 96

## L

Lee, Seungkyu 310  
Lee, Youngmoon 145  
Li, Chang 245  
Li, Frederick W. B. 262  
Li, Jingcheng 425  
Li, Ruochen 262  
Liberatori, Bendetta 327  
Liu, Meng 212  
Liu, Yang 458  
Lu, Yue 360  
Lyu, Shujing 360

## M

Mao, Yingchi 245  
Miao, Zhenjiang 392  
Misra, Debojyoti 442  
Mucha, Wiktor 178

## N

Nair, Nilah Ravi 17, 33  
Narang, Pratik 113

Narwade, Pradeep 375  
 Ngu, Anne Hee Hiong 425  
 Ni, Hao 278  
 Ni, Jianyuan 425  
 Niinuma, Koichiro 375  
 Nugent, Christopher 212

**O**

Ohara, Genji 80  
 Osman, Nada 229

**P**

Paramesran, Raveendran 343  
 Pauly, Markus 33  
 Phan, Raphaël C.-W. 343  
 Pundhir, Anshul 96

**Q**

Qian, Weiwen 245  
 Qiao, Tanqiu 262

**R**

Raman, Balasubramanian 96  
 Ramaswamy, Lakshmish 407  
 Ratha, Nalini 49, 65  
 Reining, Christopher 33  
 Ricci, Elisa 327  
 Rizvi, Syed Sameen Ahmad 113  
 Rota, Paolo 327  
 Roy, Debaditya 162  
 Rueda, Fernando Moya 17, 33

**S**

Sahu, Mihir 1  
 Sandhan, Tushar 442  
 Schmid, Lena 33  
 Seth, Aryan 113  
 Sharma, Tilak 49

Shi, Dongzi 278  
 Shum, Hubert P. H. 262  
 Singh, Arjun 1  
 Strohmayer, Julian 194

**T**

Tian, Yi 475  
 Torki, Marwan 229  
 Trehan, Shubham 293

**U**

Ubul, Kurban 129  
 Ullah, Saif 145

**W**

Wang, Fei 458  
 Wray, Michael 178

**X**

Xi, Xiaoming 212  
 Xiao, Jiahua 458  
 Xiong, Tong 278  
 Xu, Wanru 392

**Y**

Yadav, Srishti 96  
 Yadikar, Nurbiya 129  
 Yalavarthi, Bharat 49, 65  
 Yan, Yan 425  
 Yang, Haiwei 458

**Z**

Zara, Giacomo 327  
 Zhan, Hongjian 360  
 Zhang, Shibingfeng 327  
 Zhang, Xin 278  
 Zhou, Peiyong 129  
 Zou, Yishan 212