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

LNCS 15316

Pattern Recognition

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

16 Part XVI

ICPR 2024 INDIA

IAPR

Springer

MOREMEDIA

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 XVI

Springer

Editors

Apostolos Antonacopoulos
University of Salford
Salford, Lancashire, UK

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

Rama Chellappa
Johns Hopkins University
Baltimore, MD, USA

Cheng-Lin Liu
Chinese Academy of Sciences
Beijing, China

Saumik Bhattacharya
IIT Kharagpur
Kharagpur, West Bengal, India

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

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-78443-9 ISBN 978-3-031-78444-6 (eBook)
https://doi.org/10.1007/978-3-031-78444-6

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

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 IJCPR 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. In ICRP 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	Indian Statistical Institute, Kolkata, India
Josef Kittler	University of Surrey, UK
Anil Jain	Michigan State University, USA

Program Chairs

Apostolos Antonacopoulos	University of Salford, UK
Subhasis Chaudhuri	Indian Institute of Technology, Bombay, India
Rama Chellappa	Johns Hopkins University, USA
Cheng-Lin Liu	Institute of Automation, Chinese Academy of Sciences, China

Publication Chairs

Ananda S. Chowdhury	Jadavpur University, India
Wataru Ohyama	Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi	Rochester Institute of Technology, USA
Lianwen Jin	South China University of Technology, China
Laurence Likforman-Sulem	Télécom Paris, France

Workshop Chairs

P. Shivakumara	University of Salford, UK
Stephanie Schuckers	Clarkson University, USA
Jean-Marc Ogier	Université de la Rochelle, France
Prabir Bhattacharya	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 Institute of Technology, 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	Indian Institute of Technology, 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 Llados	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
Mingxin Huang
Lei Kang
Wenhui Liao
Yuliang Liu
Yongxin Shi
Da-Han Wang
Yang Xue
Wentao Yang
Jiaxin Zhang
Yiwu Zhong

Reviewers (Conference Papers)

Aakanksha Aakanksha
Aayush Singla
Abdul Muqeet
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
Anwesha 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 Gunsel
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. Martã-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 Coquenet
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
Dipanjyoti 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 Binamakhashen
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 Otroshi 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
Jiyan 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
Liviu-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 Kampel
Martina Pastorino
Marwan Torki
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
Narayan Vetrekar
Narendra D. Londhe
Nathalie Girard
Nati Ofir
Naval Kishore Mehta
Nazmul Shahadat
Neeti Narayan
Neha Bhargava
Nemanja Djuric
Newlin Shebiah R.
Ngo Ba Hung
Nhat-Tan Bui
Niaz Ahmad
Nick Theisen
Nicolas Passat
Nicolas Ragot
Nicolas Sidere
Nikolaos Mitianoudis
Nikolas Ebert
Nilah Ravi Nair
Nilesh A. Ahuja
Nilkanta Sahu
Nils Murrugarra-Llerena
Nina S. T. Hirata
Ninad Aithal
Ning Xu
Ningzhi Wang
Niraj Kumar
Nirmal S. Punjabi
Nisha Varghese
Norio Tagawa
Obaidullah Md Sk
Oguzhan Ulucan
Olfa Mechi
Oliver Tüselmann
Orazio Pontorno
Oriol Ramos Terrades
Osman Akin
Ouadi Beya
Ozge Mercanoglu Sincan
Pabitra Mitra
Padmanabha Reddy Y. C. A.
Palaash Agrawal
Palaiahnakote Shivakumara
Palash Ghosal
Pallav Dutta
Paolo Rota
Paramanand Chandramouli
Paria Mehrani
Parth Agrawal
Partha Basuchowdhuri
Patrick Horain
Pavan Kumar
Pavan Kumar Anasosalu Vasu
Pedro Castro
Peipei Li
Peipei Yang
Peisong Shen
Peiyu Li
Peng Li
Pengfei He
Pengrui Quan
Pengxin Zeng
Pengyu Yan
Peter Eisert
Petra Gomez-Krämer
Pierrick Bruneau
Ping Cao
Pingping Zhang
Pintu Kumar
Pooja Kumari
Pooja Sahani
Prabhu Prasad Dev
Pradeep Kumar
Pradeep Singh
Pranjal Sahu
Prasun Roy
Prateek Keserwani
Prateek Mittal
Praveen Kumar Chandaliya
Praveen Tirupattur
Pravin Nair
Preeti Gopal
Preety Singh
Prem Shanker Yadav
Prerana Mukherjee
Prerna A. Mishra
Prianka Dey
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
Ruowei 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
Siladittya 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
Snehasis Banerjee
Snehasis Mukherjee
Snigdha Sen
Sofia Casarin
Soheila Farokhi
Soma Bandyopadhyay
Son Minh Nguyen
Son Xuan Ha
Sonal Kumar
Sonam Gupta
Sonam Nahar
Song Ouyang
Sotiris Kotsiantis
Souhaila Djaffal
Soumen Biswas
Soumen Sinha
Soumitri Chattopadhyay
Souvik Sengupta
Spiros Kostopoulos
Sreeraj Ramachandran
Sreya Banerjee
Srikanta Pal
Srinivas Arukonda
Stephane A. Guinard
Su O. Ruan
Subhadip Basu
Subhajit Paul
Subhankar Ghosh
Subhankar Mishra
Subhankar Roy
Subhash Chandra Pal
Subhayu Ghosh
Sudip Das
Sudipta Banerjee
Suhas Pillai
Sujit Das
Sukalpa Chanda
Sukhendu Das
Suklav Ghosh
Suman K. Ghosh
Suman Samui
Sumit Mishra
Sungho Suh
Sunny Gupta
Suraj Kumar Pandey
Surendrabikram Thapa
Suresh Sundaram
Sushil Bhattacharjee
Susmita Ghosh
Swakkhar Shatabda
Syed Ms Islam
Syed Tousiful Haque
Taegyeong Lee
Taihui Li
Takashi Shibata
Takeshi Oishi
Talha Ahmad Siddiqui
Tanguy Gernot
Tangwen Qian
Tanima Bhowmik
Tanpia Tasnim
Tao Dai
Tao Hu
Tao Sun
Taoran Yi
Tapan Shah
Taveena Lotey
Teng Huang
Tengqi Ye
Teresa Alarcon
Tetsuji Ogawa
Thanh Phuong Nguyen
Thanh Tuan Nguyen
Thattapon Surasak
Thibault Napolãon
Thierry Bouwmans
Thinh Truong Huynh Nguyen
Thomas De Min
Thomas E. K. Zielke
Thomas Swearingen
Tianatahina Jimmy Francky Randrianasoa
Tianheng Cheng
Tianjiao He
Tianyi Wei
Tianyuan Zhang
Tianyue Zheng
Tiecheng Song
Tilottama Goswami
Tim Büchner

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

6-DOF Motion Blur Synthesis and Performance Evaluation of Object Detection

Hanjin Yang[1], Feng Li[1,2](✉), and Lei Zhang[2]

[1] School of Computer Science and Technology, Donghua University,
Shanghai 201600, China
lifeng@dhu.edu.cn

[2] National Innovation Center of Advanced Dyeing and Finishing Technology,
Tai'an 271000, Shandong, People's Republic of China

Abstract. The current state-of-the-art deep learning vision networks commonly employs synthetic approaches for data augmentation when confronted with scenes requiring motion blur. However, existing blur synthesis methods often fall short in accurately simulating motion blur as observed in real-world scenarios, consequently hindering the generalization capability of trained deep visual networks to real-world applications, a phenomenon known as domain shift effects [1]. To address this problem, we propose a novel non-uniform motion blur synthesis method for data augmentation. First, we randomly generate the camera's motion trajectory using a more general six-degree-of-freedom (6-DOF) camera motion model, and then map this trajectory to pixel-level blur kernels. To efficiently perform spatially varying convolution, we employ non-negative matrix factorization (NMF) to decompose those blur kernels into a set of kernel basis and their corresponding mixing coefficients. This enables parallel execution of spatial variation convolutions, thereby significantly improving the efficiency of blur synthesis. Our experiments demonstrate consistently superior results of the proposed method on publicly available real datasets RealBlur, as well as synthetic datasets GoPro and REDS.

Keywords: Motion blur synthesis · Data augmentation · 6-DOF Camera shake

1 Introduction

Motion blur synthesis [1,4,6–9] aims to transform sharp images into motion-blurred counterparts through post-processing techniques, thereby eliminating the cost of acquiring actual blurry images [1] for training deep learning visual networks. This paper focuses on the efficient simulation of spatially varying motion blur. Ideally, deep learning networks such as object detection typically operate on sharp images without any degradation. However, in real-world applications, these networks, which exhibit robust performance in controlled environments,

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 1–17, 2025.
https://doi.org/10.1007/978-3-031-78444-6_1

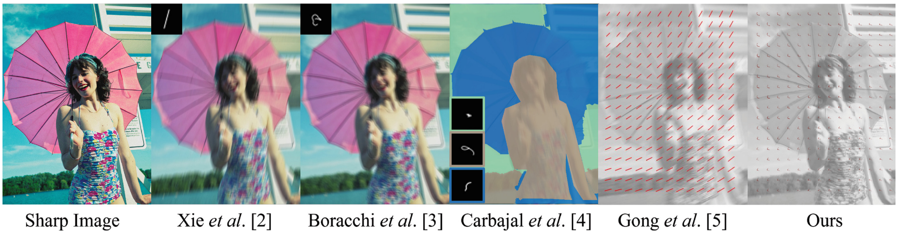


Fig. 1. Example of the blurry image synthesis by different methods. From left to right are: sharp image, blurry image synthesis by Xie et al. [2], Boracchi et al. [3], Carbajal et al. [4], Gong et al. [5] and ours, respectively. [2] and [3] are uniform motion blur synthesis methods, the rest are non-uniform. The blur kernels corresponding to these methods are plotted on the images.

frequently confront challenges posed by motion blur [2,7,8,10], leading to a decline in performance.

To enhance robustness against motion blur, existing methods adopt to incorporate blurry images into the training of neural networks. However, acquiring labeled blurry images in reality presents significant challenges [1], prompting many methods to resort to synthesizing blurry images to acquire training data. Here, “labeled” refers to the requirement of collecting additional data paired with blurry images.

Depending on whether the blur kernel is shared by all pixels in the image, motion blur can be categorized into two types : uniform motion blur and non-uniform motion blur. The simulation of uniform motion blur is relatively straightforward and allows for efficient synthesis of blurry images. Methods in [2,7,8], under the assumption of uniform blur, demonstrate strong performance in scenarios adhering to their motion blur model.

In contrast, another category of approaches assumes motion blur to be non-uniform. By benefitting from a more precise modeling of the blur kernel compared to uniform blur, methods in [4,6,11,12] achieve significant advancements in single-image deblurring tasks. However, the augmented data produced by existing non-uniform methods still encounter two primary challenges when integrated into visual systems : (1) Domain shift effects [1] : Deep learning systems trained on synthetic data often exhibit poor generalization in real-world scenarios due to the dissimilarity between the synthesized blur and real blur; (2) Computational inefficiency : Particularly when augmenting large-scale datasets like MS COCO, computational inefficiency arises primarily from the serial nature of spatial variant convolution in existing methods.

Our contributions are summarized as follows:

1. We propose an efficient non-uniform motion blur synthesis method for data augmentation in deep learning visual systems.
2. We introduce a simple yet effective method for generating random camera motion trajectory with arbitrary degrees of freedom up to six.

3. With the experimental results, we demonstrate that deep learning visual system trained with our data augmentation method outperform existing methods on various datasets.

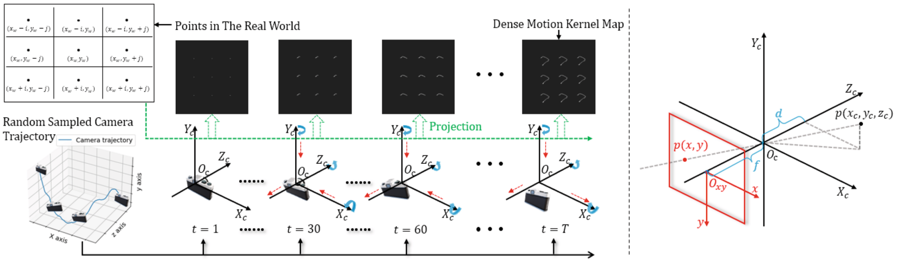


Fig. 2. On the left is our framework for generating non-uniform motion blur kernels. At each time step t, we project some points from the real world onto the image plane based on the camera pose. On the right is the pinhole camera model, $X_cY_cZ_c$ is the camera coordinate system (black), the world coordinate system coincides with it, and xy is the image plane (red) (Color figure online).

2 Related Works

2.1 Uniform Motion Blur Synthesis

In the case of uniform blur, blurring is modeled as the convolution of the sharp image with a blur kernel, the differences among various methods lie in the way the blur kernel is generated.

Some works have shown that using simple linear motion blur kernels and convolving them with sharp images can still be effective in their tasks [2,13]. However, in reality, the relative motion between the camera and the scene is often more complex and nonlinear. In order to generate more complex and realistic blur kernels, Gavant et al. [14] proposed a camera shake kernel generator based on hand physiological tremor data to obtain the blur kernel which able to mimic the mechanical response of a shoulder-arm-hand system. Boracchi et al. [3] directly considered the projection of camera motion trajectories on a two-dimensional plane. They model the motion trajectories on the plane as a Markov random process and took into account the influence of random shake when holding the camera by hand and the exposure time on the scale of the blur kernel. Due to its simple principle and low computational cost, and its ability to simulate the blur kernel (i.e., Point Spread Function, PSF) in complex camera motion scenarios, this method has been widely used.

2.2 Non-uniform Motion Blur Synthesis

In non-uniform motion blur, different positions in the image correspond to different blur kernels. Luo et al. [13] pointed out the necessity of distinguishing whether the motion occurs in the foreground or background, as failure to do so could result in pixel distortion along object edges. However, their method was proposed for controlled artistic image editing and required additional manual intervention, making it unsuitable for large-scale data generation.

To explore the performance of various blind image deblurring algorithms on real non-uniform blur images, [15] constructed a system capable of recording and replaying camera motion during photography. They recorded 40 real camera motion trajectories during exposure time and used a high-precision hexapod robot to playback these trajectories to capture real motion blur images. [5] generates non-uniform motion blur images by simulating 4-DOF camera ego-motion. They first sample a dense flow map, then pixel-wise map the optical flow to linear blur kernel parameters and perform convolution, ultimately obtaining the non-uniform blurry image. [16] considered camera motion with 6-DOF, but their method was used for light field image motion blur synthesis and cannot be used on common three-channel images. Aitor et al. [17] used image matting to segment the image to obtain masks for the regions to be blurred. After uniformly blurring the image, they combined each superpixel blocks using masks and then used inpainting to eliminate pseudo-artifacts along region boundaries. Another research [4] is similar to [17]. Instead of using inpainting, they convolved the segmented mask of the image, achieving a soft transition between different blurry regions.

Other works utilize deep learning methods instead of motion model-based approaches to generate motion blur. [18] used a pair of temporally continuous sharp images as input to obtain non-uniform motion blur images by spatially varying linear blur kernel estimation. [19] extends the implicitly encoded blur kernel to sharp images to generate motion blur images.

Recent research [20] employs neural networks to generate random but image depth-related motion offsets, and then repositions them at the pixel level according to the motion offset to generate blurry images. However, these deep learning based methods are data-driven no matter how the network structure changes. The datasets used to train these networks, such as GoPro, contain limited number of images and degradation patterns. [20] pointed out that these methods [18,19,21] although increasing the diversity of generation by sampling random noise, still have insufficient diversity in degradation patterns. Non-deep learning methods provide advantages in terms of the controllability and interpretability of blur parameters compared to deep learning methods. Therefore we utilize non-deep learning method to synthesis blurry images.

2.3 Deep Learning Visual Systems Augmented with Motion Blur

Xie et al. [2] designed a blur classification network to assist a plug-and-play module, which improves object detection on blurry images without affecting the detection performance on sharp images. Zheng et al. [10] integrated deblurring and object detection into an adversarial generative training framework, preserving the speed advantage of the object detection network while achieving motion blur robustness. They observed improvements in both blind deblurring and object detection tasks. Wang et al. [22] devised an adversarial image generation method to enhance object detection performance on open and composite degraded images. Gathering real-world composite degraded images is challenging, so they employed image-to-image translation to generate degraded images.

In [21], the author employs a real blurry dataset they collected to train a single-image deblurring network structured similarly to CycleGAN [23]. The architecture includes two generators and two discriminators. The GAN which learns blurring play a role in data augmentation to some extent, enhancing generalization on new data. Similar to [21,24] also employs a dual-branch design to learn degradation representations for single-image deblurring. They model degradation information as the residual between sharp and blurry images and optimize the degradation information representation through a deblurring branch, improving the deblurring performance.

3 Camera Shake Simulate for Blur Synthesis

3.1 Our Camera Shake Model

During the exposure time, the camera moves independently in six degrees of freedom in our camera shake model, and the light rays from the scene accumulate on the image plane, forming a motion blur image. Although the actual imaging process involves continuous motion, modeling continuous camera motion trajectories is challenging. Inspired by [3,15], we sample a series of discretized camera poses to approximate the camera motion during the exposure time.

Assuming the exposure time is T, we discretize the exposure time into M time steps. At each time step t, we generate a six-dimensional camera pose:

$$p(t) = [\theta_x(t), \theta_y(t), \theta_z(t), l_x(t), l_y(t), l_z(t)]^\top \tag{1}$$

The first three dimensions represent the rotation angles of the camera around each axis, measured in degrees. The last three dimensions represent the translations of the camera along each axis, measured in millimeters. Our coordinate system is shown on the right side of Fig. 2. Each trajectory consists of M positions generated by a particle moving randomly in a six-dimensional continuous space. The process of trajectory generation is detailed in Algorithm 1 (at the end of the manuscript) . The parameter settings initially follows a similar algorithm [3] and then adjusts to fit the data range of real-recorded trajectories from [15].

3.2 6-DOF Trajectory Transform to Dense Motion Kernel Map

Mapping of Real-World Points to the Image Plane. The 6-DOF trajectory records camera poses at continuous time steps. To convert the trajectory into a motion blur kernel, we need to utilize the pinhole camera model to map points in the real world coordinate system to the pixel coordinate system. By repeating this operation for each time step of a trajectory, we can obtain a series of projected points on the pixel coordinate system, as shown in Fig. 2. The trajectory formed by these points constitutes the blur kernel, also known as the Point Spread Function (PSF).

Without considering camera lens distortion, for a point $(x_w, y_w, z_w)^\top$ in the world coordinate system , its coordinates in the image plane can be obtain by:

$$\begin{bmatrix} u \; v \; 1 \end{bmatrix}^\top = \frac{1}{d} \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0_3^\top & 1 \end{bmatrix} \begin{bmatrix} x_w \; y_w \; z_w \; 1 \end{bmatrix}^\top \tag{2}$$

The two matrices multiplied by the point $(x_w, y_w, z_w)^\top$ are respectively the camera's extrinsic matrix and intrinsic matrix, d represents the distance from the point to the X_cY_c plane, it is also equal to z_w, as shown on the right side of Fig. 2. The camera's intrinsic parameters can be obtained through calibration methods [25]. In this paper, we directly utilize the calibrated parameters from [15]. The rotation matrix R and translation matrix t in the extrinsic matrix vary with the camera's pose change, the two matrix can be derived from the rotation and translation components of the camera pose in Eq. 1 respectively.

According to Eq. (2), a point in the real world can be mapped to the image plane. For a camera motion trajectory with M time steps, performing M mappings yields M points on the image plane. Since the differences in camera poses between each pair of adjacent time steps are small, these points naturally connect without interruption, forming a trajectory on the plane. Rotating the trajectory on the image plane by 180° (due to the pinhole camera model forming an inverted image) yields a blur kernel.

Furthermore, to obtain the Dense motion kernel Map based on the camera motion trajectory, we only need to sample multiple points in the real world, and then project these points onto the image plane at once. We noticed that very few sampling points (far fewer than the number of pixels in the image) can still yield good results. More details can be found in Sect. 4.4.

Selection of Points in the Real-World. Our objective is to synthesize non-uniform motion blur, which requires sampling multiple points according to certain rules to obtain a set of spatially related PSFs, forming a Dense motion kernel Map. In our method, we only consider motion blur caused by camera ego-motion (i.e., only the camera is moving while objects in the scene remain stationary). It is easy to demonstrate that when the camera pose is fixed, all points along the ray O_cP will map to the same location on the image plane, as shown in Fig. 2. Therefore, instead of considering all points in the world coordinate system, we can use a bounded (due to the fixed field of view of the camera being less than 180°in both horizontal and vertical directions) plane that is parallel to the image plane and a distance d away from the X_cY_c plane. We refer to this plane as $Plane_d$. Intuitively, this plane can be thought of as blocking all light rays passing through the pinhole imaging model. We only need to sample the two-dimensional coordinates (x_w, y_w) of points on this plane because the points on $Plane_d$ share the same z coordinate. In this paper, the distance d between the object and the camera is set to 62 cm, which is consistent with [15], in this way, the parameter adjustment of our trajectory generation algorithm can be based on the real-recorded trajectories from [15].

Specifically, we take the origin of $Plane_d$ (i.e. the intersection of the z axis and $Plane_d$) as the center, and perform equal-interval sampling in the x and y directions to obtain $P * P$ points. The equal interval ensures the representativeness of these PSFs obtained from point mapping, and sampling the same number of steps in both directions is for convenience.

3.3 Efficient Blur Synthesis

The entire process of blur synthesis is primarily time-consuming in two aspects: the generation of heterogeneous kernel maps and the convolution of kernels with the image. Our method inherently benefits from parallel generation of the Dense motion kernel Map, in this way, the key factor affecting the speed lies in the convolution process.

Performing pixel-wise convolution in a serial manner is time-consuming, inspired by [4], we implement efficient non-uniform blur synthesis using an adaptive basis decomposition approach, preserving the complexity of pixel level blur kernels while allowing parallelization of the blurring process. Specifically, based on the assumption that pixel level blur kernels exhibit redundancy [4], we decomposing the kernel Map into a linear combination of basis blur kernels. In our framework, given a sharp image S, the blurry image is generated as:

$$B = \sum_{b=1}^{C} S * \mathbf{K}^b m^b + N \tag{3}$$

where $*$ denotes the convolution, C is the number of basis blur kernels and we set $C = 25$ follow [4] , $\mathbf{K}^b$ is the b-th basis blur kernel, m^b is the pixel level mixing coefficient matrix corresponding to the b-th basis, and N is additive noise.

Decomposing the Dense motion kernel Map needs to satisfy two fundamental conditions: firstly, the basic properties of acknowledged blur kernels, namely non-negativity and normalization; secondly, the mixing coefficients need to be non-negative and normalized to achieve convex combinations of the basis blur kernels, thereby ensuring energy conservation.

To meet the aforementioned conditions, we opt for Non-negative Matrix Factorization (NMF) as the decomposition method. NMF is capable of decomposing a non-negative matrix into the product of two non-negative matrices:

$$M_{m\times n} \approx K_{m\times k} W_{k\times n} \tag{4}$$

where $M_{m\times n}$ is the Dense motion kernel Map, m and n are the dimension of a single PSF and the number of sampling points respectively. $K_{m\times k}$ is the k basic blur kernels with dimension m, corresponding to $\mathbf{K}$ in Eq. 3. $W_{k\times n}$ represents k mixing coefficient matrices with dimension n, to extend to pixel-level non-uniform blurring, we simply upsample each mixing coefficient matrix of dimension n to the size of the image, thereby obtaining the pixel-level m in Eq. 3. For normalization, we normalize the obtained $K_{m\times k}$ and $W_{k\times n}$ respectively after NMF. Examples of non-uniform blurring of image and decomposition of kernel map are shown in Fig. 3.

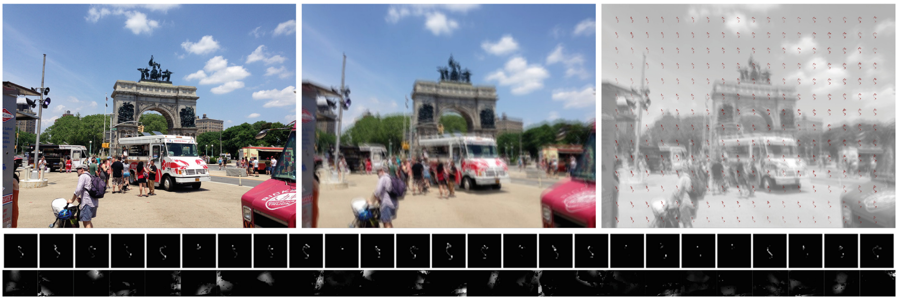


Fig. 3. Example of adaptive decomposition of Dense motion kernel Map. The images in the top row from left to right are, sharp image, synthetic blurry image, and visualized kernel map. The following two lines are the decomposed kernel basis and corresponding mixing coefficients respectively.

4 Experiments

Our non-uniform motion blur synthesis method was proposed to mitigate the domain shift effect [22] introduced during data augmentation. The domain shift problem manifests when deep learning networks trained on synthetic data perform well on synthetic data but exhibit performance degradation on real data. To demonstrate the effectiveness of our blur synthesis method, we choose the representative object detection network YOLOv7 [26] from the deep learning visual networks as our experimental platform.

We compare our blur synthesis method with another four blur synthesis methods. Specifically, we applied each synthesis method to generate motion-blurred training data from sharp images in the MS COCO [27] dataset. We then train the object detection network using this data and evaluate the performance of object detection on a real blur dataset **RealBlur** [1], as well as two synthetic blur datasets **GoPro** [28] and **REDS** [29]. The better performance of object detection indicates that the synthesized blur is closer to the real blur because the domain shift is smaller.

4.1 Datasets

COCO2017 Dataset. To ensure the object categories of the three deblurring datasets as inclusive as possible, we choose the COCO2017 dataset [27] with 80 object categories as the source of sharp images,which involves a total of 123287 (merged the train and val sets) images. Each blur synthesis method utilize the sharp images in COCO2017 to generate blurry images.

Three Deblurring Datasets. The **RealBlur** dataset [1] is the only dataset among these three deblurring datasets that contains real blur images. **Gopro** [28] and **REDS** [29], are both perform inter-frame averaging on high frame rate videos to obtain blurry images. These datasets were collected for deblurring tasks and without bounding box labels. We employs the YOLOv8 [30] to perform object detection on the sharp images in these three datasets (all data, including training and test sets) , and the detection results were used as ground truth bounding boxes (GT Box).

4.2 Implementation details

Blur Synthesis Parameter Settings. The naming and parameter settings of each synthesis methods are as follows: (a) *Purelin* [2] : Linear blur kernels are generate using the MATLAB toolbox, with length and angles ranging from [1,33] and [0,360] uniformly; (b) *Markov* [3] : The parameter settings followed [7] ; (c) *EFF* [4] : The parameter settings followed [4] ; (d) *Denselin* [5] : The parameter settings followed [5]. The blur kernels generated by each method were centered as described in [7] to reduce positional discrepancies between the data and labels.

Experimental Setup. We compare the performance of different motion blur synthesis methods on the object detection network YOLOv7 [26]. For training YOLOv7, we fine-tuning the pretrained weights on sharp images instead of training from scratch. The same settings are applied to each synthesis method : pretrained weights were used, training for 100 epochs with a batch size of 16, and other settings are consistent with the original training setup of original YOLOv7, including random flipping and Mixup [31] augmentation.

4.3 Performance Comparison

In this section, we discuss the performance of the object detection network trained with different motion blur synthesis methods on real and synthetic blurry datasets. To evaluate the performance of object detection, we employ the commonly used AP (Average Precision) and mAP (mean of Average Precision) metrics mAP@0.5 and mAP@0.5:0.95 [27]. In our experiments, the higher these two metrics, the smaller the domain shift effect and the synthesized blur is closer to the real blur, we label these two metrics for detecting blurry images in three datasets as mAP_{50} and $\text{mAP}_{50:95}$ respectively. To reduce the impact of randomness, each blur synthesis method generates three batches of training data, and the reported results are the average mAP over three training runs.

Table 1. AP for YOLOv7 trained on COCO2017 images with synthesized blur and evaluated on Gopro [28]. The last two columns are mAP for all categories.

Method	Bench	Bicycle	Motorcycle	Person	PottedPlant	Truck	mAP$_{50}$	mAP$_{50:95}$
Original	65.85	63.26	53.43	87.14	49.26	57.78	64.37	45.12
Purelin[2]	58.75	76.64	61.84	91.50	74.40	80.37	70.75	49.52
Markov[3]	65.34	77.20	66.72	91.15	69.30	79.56	71.63	49.65
EFF[4]	69.32	77.94	59.89	90.81	72.72	77.97	70.51	49.35
Denselin[5]	63.27	76.72	62.94	91.17	75.20	80.24	70.22	48.87
Proposed	70.29	76.02	63.85	91.49	75.22	81.00	73.87	51.47

Table 2. AP for YOLOv7 trained on COCO2017 images with synthesized blur and evaluated on RealBlur [1]. The last two columns are mAP for all categories.

Method	Bench	Bicycle	Bottle	Bowl	PottedPlant	Truck	Umbrella	mAP$_{50}$	mAP$_{50:95}$
Original	69.70	78.81	31.47	67.24	63.27	65.95	68.73	64.30	51.77
Purelin[2]	77.63	85.51	36.61	75.38	71.69	59.06	80.58	66.41	51.35
Markov[3]	77.41	86.03	46.73	76.72	69.81	65.38	76.48	66.63	51.56
EFF[4]	74.65	86.91	34.45	71.54	65.26	65.11	76.86	65.43	50.78
Denselin[5]	74.54	88.38	34.51	74.77	68.73	64.53	74.68	65.43	50.10
Proposed	79.54	89.04	43.93	77.86	67.13	66.68	83.19	68.48	53.75

Table 3. AP for YOLOv7 trained on COCO2017 images with synthesized blur and evaluated on REDS [29]. The last two columns are mAP for all categories.

Method	Boat	Car	Chair	Person	PottedPlant	Truck	Umbrella	mAP$_{50}$	mAP$_{50:95}$
Original	57.86	83.53	47.95	84.49	60.69	66.17	61.47	65.49	45.68
Purelin[2]	64.75	86.79	61.65	89.37	66.71	67.92	60.95	68.11	46.83
Markov[3]	61.67	87.56	60.45	89.27	69.67	70.28	60.00	68.70	47.37
EFF[4]	61.07	86.41	60.29	88.64	66.87	67.92	59.09	67.51	45.98
Denselin[5]	60.75	85.93	61.26	89.04	66.78	66.81	60.88	67.22	45.49
Proposed	62.17	87.96	62.92	90.31	69.28	71.10	62.77	69.28	47.91

Tables 1, 2 and 3 presents the testing results of the object detection network YOLOv7 [26] trained with different augmentation methods on the **Gopro**, **RealBlur** and **REDS** datasets, respectively. Red and blue colors are used to indicate the 1st and 2nd ranks, respectively. Limited by the width of the table, we only show the AP values of some object categories with the largest differences, while the mAP value is for all object categories in each dataset.

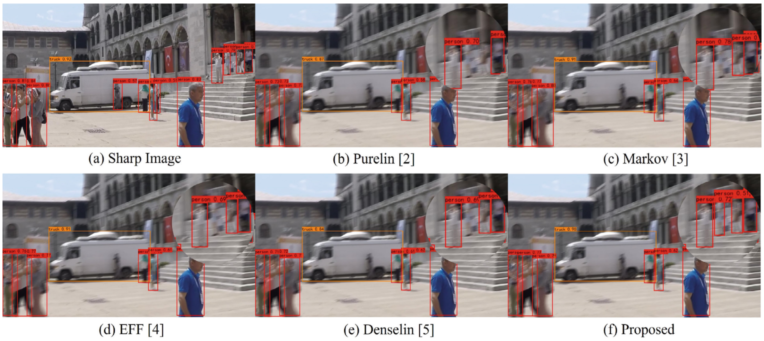
As expected, the detection performance improves on blurry images after finetuning with all synthesis method. From the results, the performance of *Markov* method [3] is better than *Purelin* [2] on three datasets. Both methods are uniform and the entire image shares a blur kernel. Before the *Markov* method [3], the *Purelin* method was widely used for motion blur synthesis. [3] can simulate motion blur more realistically than linear blur method *Purelin* [2] and is computationally simple.

Unexpectedly, the non-uniform motion blur synthesis methods *EFF* [4] and *Denselin* [5] perform worse than the two uniform blur synthesis methods *Purelin* [2] and *Markov* [3]. The results on three datasets indicate that the performance of these two non-uniform motion blur synthesis methods is similar, but both are inferior to the uniform blur methods, which indicates that the blurry images produce by these two methods are far from the real blur.

We speculate that the poor performance of *EFF* [4] is due to distortion when handling the boundaries of different blurry regions. Although their soft transition strategy is visually artifact-free, the blur kernel at the junction of different regions is actually mathematically unknown, and the results in the above tables indicate that this unrealistic transformation increases the domain shift effect.

For *Denselin* [5], the training results in Table 2 indicate a significant discrepancy between their method and real motion blur. We speculate that their poor performance was due to their method only being able to simulate the motion of the camera's four degrees of freedom, leading to deviations in motion patterns. Further experiments are detailed in Sect. 4.4.

From the results in Tables 1, 2 and 3, our method achieved the best performance (mAP) on all three datasets. Compared to the two uniform blur synthesis methods *Purelin* [2] and *Markov* [3], our method is non-uniform. Compared to the two non-uniform blur synthesis methods *EFF* [4] and *Denselin* [5], our blur kernel is non-linear. This implies that non-uniform and non-linear motion blur synthesis method is able to generate data closer to real-world motion blur, thereby reducing the impact of domain shift at the data level. The improvement in object detection with various augment methods under blurred scenes is depicted in Fig. 4.


(a) Sharp Image

(b) Purelin [2]

(c) Markov [3]

(d) EFF [4]

(e) Denselin [5]

(f) Proposed

Fig. 4. Example of the effectiveness of various augmentation methods in training object detection network. In this image form **Gopro** [28], only our method can detect all people in the upper right corner. We enlarged a portion of each image for better observation.

4.4 Ablation study

In this section, we conduct ablation experiments on each component of the proposed non-uniform motion blur synthesis framework.

The Freedom of Camera Motion. To validate the effectiveness of full camera motion freedom(i.e. 6-DOF) on motion blur synthesis, we modified the non-uniform blur synthesis method *Denselin* [5], which initially had only 4-DOF, to simulate 6-DOF motion. We then compared it with our method.

Specifically, we retain the characteristic of pixel-level linear kernels and re-generate training data using a new optical flow generation method [32] with 6-DOF, the transformation from optical flow to pixel-level linear kernels remains consistent with the original *Denselin*, then retrain the object detection network with the re-generated data. [32] simulates arbitrary camera motion in 3D space to generate optical flow for a single image. The re-generated data labeled as *Denselin6D*.

Table 4. The mAP@0.5 for the proposed method and two variations of the *Denselin* method on three datasets. Best results are shown in bold.

Methods	DOF	BlurKernel	Gopro[28]	RealBlur[1]	REDS[29]
Denselin[5]	4	linear	70.22	65.06	67.22
Denselin6D	6	linear	71.24	66.97	68.57
Proposed	6	non-linear	**73.87**	**68.48**	**69.28**

Table 4 presents a comparison of two *Denselin* [5] methods using different optical flow strategies. From the results, the *Denselin6D* with more degrees of freedom shows an improvement compared to *Denselin*, but still slightly worse than the uniform blur method *markov* [3] on the **Gopro** and **REDS** datasets, refer to Table 1, which means that, besides the degrees of freedom of camera motion, the linear kernel is also an important factor limiting the performance of *Denselin*. Our method and *Denselin6D* are both capable of simulating 6-DOF camera motion blur. However, the difference lies in the fact that the pixel-level blur kernel in *Denselin6D* is linear, whereas our is non-linear, linear kernel is just a special case in our method. The Dense blur kernel corresponding to these two methods are shown in Fig. 1.

Our camera motion model assumes that the camera's motion in the six degrees of freedom is independent, allowing our method to simulate random camera motion with up to six degrees of freedom. To demonstrate the backward compatibility of our method, we freeze certain degrees of freedom of camera motion, and the resulting Dense motion kernel Maps are shown in Fig. 5.

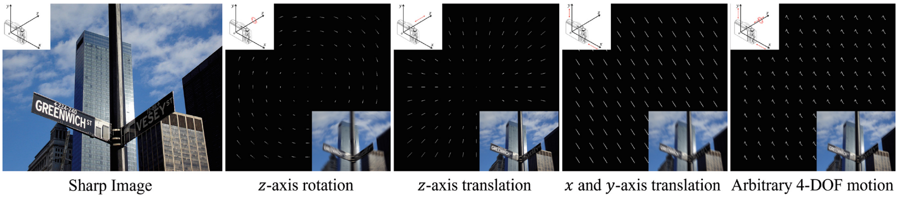


Fig. 5. Demonstration of the backward compatibility of our method. The Dense motion kernel Map is depict in the figure, with the corresponding camera motion pattern and blurry image shown in the top left and bottom right corners, respectively. Best viewed when zoomed in.

The Other Parameters in Our Method. Our method involves two another primary parameters : camera impulsive shake probability p_s and the number of sampled points num. The former parameter controls the degree of camera impulsive shake, a larger value resulting in more distorted blur kernels. The latter parameter controls the degree of non-uniformity of the kernel map, with larger values resulting in greater variability of the kernel across different positions in the image.

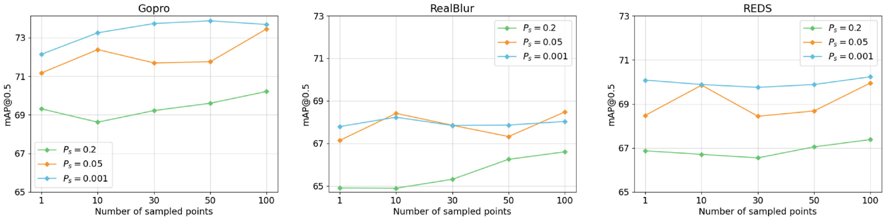


Fig. 6. Evaluation of the augmentation performance of the object detection network with data generated under different parameter combinations. The sampling points in the legend correspond to the square of the actual number.

Table 5. The average time taken by each method to synthesis a single blurry image. The first three columns represent our method.

methods	10 × 10	50 × 50	100 × 100	Purelin[2]	Markov[3]	EFF[4]	Denselin[5]
times/s	0.12	0.29	1.01	0.03	0.17	0.29	0.68
non-uniform	✓	✓	✓	✗	✗	✓	✓

Figure 6 illustrates the effectiveness of the generated data under different parameter settings, where higher mAP values indicate a closer resemblance between the synthesized blur and real blur. From the results, the overall trend on the three datasets indicates that more sampling points, i.e., higher non-uniformity of the blur, result in blur that is closer to reality. The best performance was achieved with kernels close but not entirely linear and with a higher number of sampling points. However, as the number of sampling points increases, the computational cost also increases. More specific speed comparisons can be found in Table 5. We observed that a satisfactory balance between speed and performance could be achieved when the number of sampling points is 10 × 10.

Figure 7 explains the non-uniformity of the Dense motion kernel Map when our method has different sampling points $num * num$. Since we utilize a fixed interval to sample points on $Plane_d$, the more sampling points there are, the farther the edge points are from the center, and the more non-uniform the PSFs generated by the projection are.

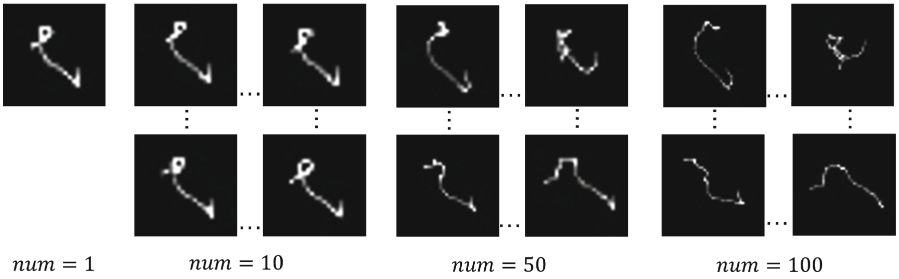


Fig. 7. The non-uniformity of the Dense motion kernel Map corresponding to the same trajectory under different numbers of sampling points. For ease of observation, we only display the kernels at the four corners of the kernel map.

Algorithm 1. 6-DOF Trajectory generation.

Parameters:

$M = 167$ - number of iterations,

$L_{rot} = 0.5$ - max length of the rotation component movement,

$L_{shift} = 5$ - max length of the translation component movement,

p_s - probability of impulsive shake, uniform from (0,0.2),

I - inertia term, uniform from (0,0.7),

P_b - probability of big shake, uniform from (0,0.1),

P_g - probability of gaussian shake, uniform from (0,0.7),

v - velocity of particle,with six dimensions, the superscript acts the same as x.

x - trajectory vector, with six dimension, the first and last three dimension labeled as $x^{'}$ and $x^{''}$ respectively.

```
procedure $Generate6DTrajectory(M, p_s, p_b, p_g)$
    $v_0^{'} \leftarrow randn(3); v_0^{''} \leftarrow randn(3)$
    $v^{'} \leftarrow v_0^{'} * L_{rot}/(M-1); v^{''} \leftarrow v_0^{''} * L_{shift}/(M-1)$
    $x = \text{zeros}(M, 6)$
    for $t = 1$ to $M - 1$ do
        if randn $< p_s * p_b$ then
            nextDir $\leftarrow 2 * v * sin(\pi + \text{randn} - 0.5)$
        else
            nextDir $\leftarrow$ zeros(6)
        $dv^{'} \leftarrow \text{nextDir} + p_s * (p_g * \text{randn}(3) - I * x^{'}[t]) * (L_{rot}/(M-1))$
        $dv^{''} \leftarrow \text{nextDir} + p_s * (p_g * \text{randn}(3) - I * x^{''}[t]) * (L_{shift}/(M-1))$
        $v^{'} \leftarrow v^{'} + dv^{'}; v^{''} \leftarrow v^{''} + dv^{''}$
        $v^{'} \leftarrow (v^{'}/\|v^{'}\|) * (L_{rot}/(M-1)); v^{''} \leftarrow (v^{''}/\|v^{''}\|) * (L_{shift}/(M-1))$
        $x[t+1] \leftarrow x[t] + v$
    return $x$
```

5 Conclusion

In this study, we propose a novel method for synthesizing non-uniform motion blur in single images, aimed at augmenting data for deep learning vision networks. With the proposed motion blur framework, we are able to synthesis more accurate pixel-level non-linear motion blur. Experimental results on both synthesized and real data demonstrate the superiority of our proposed method.

References

1. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, pp. 184–201. Springer (2020)
2. Xie, G., Li, Z., Bhattacharyya, S., Mehmood, A.: Plug-and-play deblurring for robust object detection. In: 2021 International Conference on Visual Communications and Image Processing (VCIP), pp. 1–5. IEEE (2021)
3. Boracchi, G., Foi, A.: Modeling the performance of image restoration from motion blur. IEEE Trans. Image Process. **21**(8), 3502–3517 (2012)
4. Carbajal, G., Vitoria, P., Delbracio, M., Musé, P., Lezama, J.: Non-uniform Blur Kernel Estimation Via Adaptive Basis Decomposition (2021). arXiv preprint arXiv:2102.01026
5. Gong, D., et al.: From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2319–2328 (2017)
6. Schmidt, U., Jancsary, J., Nowozin, S., Roth, S., Rother, C.: Cascades of regression tree fields for image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **38**(4), 677–689 (2015)
7. Sayed, M., Brostow, G.: Improved handling of motion blur in online object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1706–1716 (2021)
8. Cho, S.J., Kim, S.W., Jung, S.W., Ko, S.J.: Blur-robust object detection using feature-level deblurring via self-guided knowledge distillation. IEEE Access **10**, 79491–79501 (2022)
9. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)
10. Zheng, S., Wu, Y., Jiang, S., Lu, C., Gupta, G.: Deblur-yolo: real-time object detection with efficient blind motion deblurring. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
11. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. Int. J. Comput. Vis. **98**, 168–186 (2012)
12. Ji, X., Wang, Z., Satoh, S., Zheng, Y.: Single image deblurring with row-dependent blur magnitude. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12269–12280 (2023)
13. Luo, X., Salamon, N.Z., Eisemann, E.: Controllable motion-blur effects in still images. IEEE Trans. Visual Comput. Graphics **26**(7), 2362–2372 (2018)

14. Gavant, F., Alacoque, L., Dupret, A., David, D.: A physiological camera shake model for image stabilization systems. In: SENSORS, 2011 IEEE, pp. 1461–1464. IEEE (2011)
15. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part VII 12, pp. 27–40. Springer (2012)
16. Lumentut, J.S., Williem, Park, I.K.: 6-DoF motion blur synthesis and performance evaluation of light field deblurring. Multimed. Tools Appl. **78**(23), 33723–33746 (2019)
17. Alvarez-Gila, A., Galdran, A., Garrote, E., Van de Weijer, J.: Self-supervised blur detection from synthetically blurred scenes. Image Vis. Comput. **92**, 103804 (2019)
18. Brooks, T., Barron, J.T.: Learning to synthesize motion blur. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6840–6848 (2019)
19. Tran, P., Tran, A.T., Phung, Q., Hoai, M.: Explore image deblurring via encoded blur kernel space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11956–11965 (2021)
20. Jing, Z., Zhang, Y., Wang, C., Liu, D., Xia, Y.: Semantically-consistent dynamic blurry image generation for image deblurring. In: Proceedings of the 30th ACM International Conference on Multimedia, p. 25472555. MM ’22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3503161.3548106,
21. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2737–2746 (2020)
22. Wang, W., Zhang, J., Zhai, W., Cao, Y., Tao, D.: Robust object detection via adversarial novel style exploration. IEEE Trans. Image Process. **31**, 1949–1962 (2022)
23. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
24. Li, D., Zhang, Y., Cheung, K.C., Wang, X., Qin, H., Li, H.: Learning degradation representations for image deblurring. In: European Conference on Computer Vision, pp. 736–753. Springer (2022)
25. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
26. 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)
27. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer (2014)
28. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3883–3891 (2017)
29. Nah, S., et al.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0 (2019)

30. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (2023). https://github.com/ultralytics/ultralytics
31. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond Empirical Risk Minimization. arXiv preprint arXiv:1710.09412 (2017)
32. Aleotti, F., Poggi, M., Mattoccia, S.: Learning optical flow from still images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15201–15211 (2021)

OVOSE: Open-Vocabulary Semantic Segmentation in Event-Based Cameras

Muhammad Rameez Ur Rahman[1](✉), Jhony H. Giraldo[2], Indro Spinelli[1],
Stéphane Lathuilière[2], and Fabio Galasso[1]

[1] Sapienza University of Rome, Rome, Italy
{rahman,spinelli,galasso}@di.uniroma1.it

[2] LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
{jhony.giraldo,stephane.lathuiliere}@telecom-paris.fr

Abstract. Event cameras, known for low-latency operation and superior performance in challenging lighting conditions, are suitable for sensitive computer vision tasks such as semantic segmentation in autonomous driving. However, challenges arise due to limited event-based data and the absence of large-scale segmentation benchmarks. Current works are confined to closed-set semantic segmentation, limiting their adaptability to other applications. In this paper, we introduce OVOSE, the first Open-Vocabulary Semantic Segmentation algorithm for Event cameras. OVOSE leverages synthetic event data and knowledge distillation from a pre-trained image-based foundation model to an event-based counterpart, effectively preserving spatial context and transferring open-vocabulary semantic segmentation capabilities. We evaluate the performance of OVOSE on two driving semantic segmentation datasets DDD17, and DSEC-Semantic, comparing it with existing conventional image open-vocabulary models adapted for event-based data. Similarly, we compare OVOSE with state-of-the-art methods designed for closed-set settings in unsupervised domain adaptation for event-based semantic segmentation. OVOSE demonstrates superior performance, showcasing its potential for real-world applications. The code is available at https://github.com/ram95d/OVOSE.

Keywords: Open vocabulary segmentation · Low-level vision · Distillation

1 Introduction

Event cameras, known for their exceptional temporal resolution, low latency, and motion blur resistance, have transformed various deep learning applications [10]. Use cases include autonomous driving [4], object recognition [8], and semantic segmentation [2]. Their outstanding performance in challenging conditions makes

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_2.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 18–33, 2025.
https://doi.org/10.1007/978-3-031-78444-6_2

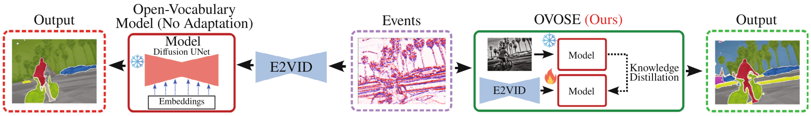


Fig. 1. Output of a regular RGB foundation model for semantic segmentation and OVOSE in event-based data. OVOSE accurately segments person, trees, and (the sky)

event cameras optimal for capturing reliable visual data [19]. Despite their success, integrating event cameras into existing computer vision models is challenging. Their unique data format, featuring asynchronous event streams without traditional image frames [28], necessitates a reassessment of established techniques. While traditional image-based semantic segmentation has made notable progress [17], event cameras, being less prevalent in real-world scenarios, suffer from a scarcity of both raw and labelled data. This fact raises two intertwined issues: the impossibility of collecting internet-scale event datasets and, as a consequence, the difficulty in training data-intensive deep learning techniques. This challenge obstructs effective semantic segmentation model training, especially in scenarios lacking established benchmarks or methodologies.

Recently, powerful foundation models have demonstrated their effectiveness in segmenting conventional images [17]. Some of these models extend the closed-set capabilities of classical semantic segmentation models [5,6], performing well in open-vocabulary settings where the set of classes in training and testing are disjoint. However, applying these powerful open-vocabulary models directly to event-based images is challenging, and retraining them is impractical due to the scarcity of annotated event-based data [15].

The alternative approach of converting events into images using E2VID [25], and pairing it with an open-vocabulary model does not perform satisfactorily as illustrated in Fig. 1. Despite the fact that E2VID is intended to reduce the domain gap between images and events, there still exists a difference between real grayscale and reconstructed-event images. In this paper, we introduce OVOSE, the first Open-Vocabulary Semantic Segmentation algorithm tailored for event-based data. OVOSE operates as a two-branch network, with one branch dedicated to grayscale images and the other to event data. Each branch incorporates a copy of an image foundation model, with OVOSE adapting the event branch for optimal performance in event-based data. Our algorithm integrates text-to-image diffusion [23] and a mask generator. Using a CLIP-style image encoder and MLP, we derive embeddings for conditioning the text-to-image diffusion UNet. We use UNet's features as input to the mask generator for mask generation. We categorize the mask generator's outputs using a frozen CLIP-style text encoder for open-vocabulary segmentation. To enhance model performance, we distill knowledge from the image branch to the events branch, using an E2VID model [25] for translating events to reconstructed images. OVOSE, characterized by its simplicity and effectiveness, outperforms existing models in event-based semantic segmentation, including Unsupervised Domain Adaptation (UDA) methods, demonstrating its effectiveness in addressing the open-vocabulary segmentation problem in event-based data.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we present the first open-vocabulary semantic segmentation approach tailored explicitly for event-based data.
- We introduce OVOSE that distills knowledge to transfer semantic insights from a foundation model trained on regular images to enhance open-vocabulary segmentation performance for event-based data.
- To mitigate the effects of sub-optimal reconstructions, we investigate various mask reweighting strategies and introduced a novel dissimilarity network. This network recalibrates the mask loss by leveraging the differences between reconstructed and original images, enabling precise fine-tuning of the segmentation model and thus producing robust and accurate predictions.
- We perform extensive evaluations in open-vocabulary semantic segmentation for three event datasets. OVOSE readily outperforms existing closed-set semantic segmentation methods and straightforward adaptations of open-vocabulary models. A set of ablation studies validates the key components of our algorithm.

2 Related Work

Event Camera Semantic Segmentation. Alonso et al. [1] introduced event camera semantic segmentation an Xception-type network for the DDD17 dataset but suffered from limitations in the quality of generated labels. Gehrig et al. [11] improved performance substantially by utilizing a synthetic event dataset converted from videos. Wang et al. [30] explored knowledge distillation and transfer learning between images and events, though relying on labeled datasets. Messikommer et al. [20] proposed a method aligning image and event embeddings but faced challenges with hallucinations [28]. Sun et al. [28] addressed domain gap reduction between events and images, yet required real event-based data. Yang et al. [35] presented a self-supervised learning framework, but still relied on labeled datasets for fine-tuning. While existing methods are designed for closed-set semantic segmentation with limited known classes, our approach is an open-vocabulary segmentation method tailored for event-based cameras, trained solely on synthetic unlabeled datasets.

Open-Vocabulary Semantic Segmentation. Recent approaches in open-vocabulary semantic segmentation for regular images have centered on embedding spaces linking image pixels to class descriptors [3,32]. Some methods leverage CLIP [22] for text and image embeddings [16], while others combine CLIP with Vision Transformer [9]. OpenSeed is introduced [36] for joint segmentation and detection tasks. ODISE [34] uniquely merges pre-trained text-image diffusion and discriminative models [26], excelling in open-vocabulary panoptic segmentation. However, these methods are designed for image data and don't directly translate to event cameras due to the fundamentally different data representation (continuous event streams vs. static images). This work proposes a novel approach for open-vocabulary semantic segmentation on event cameras. We bridge the gap between image-based methods and event data by transferring knowledge from a powerful image foundation model to a new model specifically tailored for the event domain.

2.1 Knowledge Distillation

Prior knowledge distillation (KD) [14] methods focus on single modality, *i.e*, image using logits [33] or features [21] or across different modalites [37] utilizing paired data. KD is also applied to event cameras [31] to distill knowledge from the image-based teacher model to the event-based student model. However, they require labeled image datasets and unlabelled real events and frames for effective transfer in semantic segmentation. Furthermore, [30] employs event-to-image transfer for semantic segmentation, but it gives poor performance [28] when applied for semantic segmentation. Unlike these approaches, we employ a synthetic training dataset and distill knowledge from an image foundation model to a foundation model tailored for open vocabulary semantic segmentation in events. The problem becomes complex as we strive to bridge the gap between events and images, and additionally tackle the synthetic-to-real gap.

3 Open-Vocabulary Segmentation in Events

3.1 Preliminaries

Event Representation. Each pixel in an event camera operates independently and reports brightness changes asynchronously, signaling only when the changes exceed a certain threshold. When a change is detected, an event is generated, capturing the pixel positions (x_i, y_i), timestamp (t_i), and polarity (p_i), indicating whether a change involves an increase or decrease in brightness. Consequently, each event can be represented as $\mathbf{E} = [x_i, y_i, t_i, p_i]$. In this paper, we transform events into grid-like representations [12], such as voxel grid [28,38] to facilitate further processing. We utilize a voxel grid representation of events as input to our model.

Text-to-Image Diffusion UNet. The text-to-image diffusion model [23] generates high-quality images from textual descriptions, utilizing the power of pre-trained encoders like CLIP [22] to encode text into embeddings. The process starts by adding Gaussian noise to images, and then the UNet architecture effectively reverses this noise, guided by cross-attention mechanisms that align the text embeddings with visual features, making the features semantically rich. We incorporated the text-to-image diffusion UNet into our model to extract rich features that are relevant to the text.

Problem Statement. Open vocabulary represents a generalization of the zero-shot task in semantic segmentation. In this setting, a model predicts masks for unseen classes $\mathcal{C}^{unseen}$ by learning from labeled data of seen classes $\mathcal{C}^{seen}$. The sets of seen and unseen classes are separate and do not overlap, *i.e*, $\mathcal{C}^{unseen} \cap \mathcal{C}^{seen} = \emptyset$. The objective of this work is to train a model $F_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ to predict the segmentation map of some stream of events. To solve this, we have a unlabeled

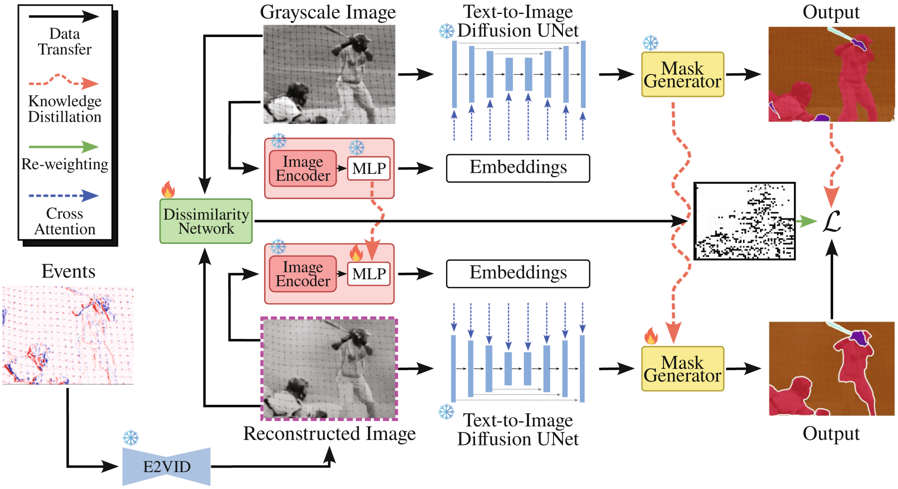


Fig. 2. Overview of OVOSE pipeline. Our algorithm comprises two components: the original grayscale image branch and the event-based branch. Initially, events are transformed into a grayscale image using the E2VID model. Subsequently, both the original and reconstructed grayscale images undergo text embedding through an image encoder and an MLP. The features from a frozen text-to-image diffusion UNet are then extracted for each tuple of image and text embedding. For each branch, a mask generator predicts class-agnostic binary masks and associated mask embedding features. Categorization is achieved through a dot product between mask embedding features and text embeddings. Both branches are initialized with ODISE weights [34], and knowledge distillation occurs from the original image branch to the event-based branch during training. Original and reconstructed images are input into a dissimilarity network to weigh the distillation in the outputs. During the evaluation, only the event-based branch is utilized.

training set $\mathcal{X}_{train} = \{\mathbf{E}_i^{(t)}, \mathbf{X}_i^{(t)}\}_{i=1}^{N_t}$, where $\mathbf{E}_i^{(t)}$ is the ith stream of events, $\mathbf{X}_i^{(t)} \in \mathbb{R}^{H \times W}$ is the ith original grayscale image with H and W the height and width of the image, and N_t is the number of streams in the dataset. For the training set, we additionally need the set of seen classes $\mathcal{C}^{seen}$. We evaluate $F_{\boldsymbol{\theta}}$ in a testing set $\mathcal{X}_{test} = \{\mathbf{E}_i^{(s)}, \mathbf{Y}_i^{(s)}\}_{i=1}^{N_s}$ where $\mathbf{Y}_i^{(s)} \in \{0,1\}^{H \times W \times |\mathcal{C}^{unseen}|}$.

3.2 Overview of OVOSE

As shown in Fig. 2 OVOSE is divided into two sections: (i) the image branch that takes as input the original grayscale images $\mathbf{X}_i^{(t)} \in \mathcal{X}_{train}$, and (ii) the event branch that takes as input the stream of events $\mathbf{E}_i^{(t)} \in \mathcal{X}_{train}$ or $\mathbf{E}_i^{(s)} \in \mathcal{X}_{test}$. It is worth clarifying that during the evaluation we only use the event branch. During training, the whole image branch is frozen and only used to distill knowledge to the event branch. For the event branch, we use the pre-trained E2VID model with parameters $\boldsymbol{\theta}_e$ [25] to transform some stream of events $\mathbf{E}$ into a reconstructed image $\hat{\mathbf{X}}$, thus $F_{\boldsymbol{\theta}_e}(\mathbf{E}) = \hat{\mathbf{X}}$. E2VID introduces a novel approach by leveraging a convolutional recurrent neural network architecture to process event camera data's sparse and asynchronous nature, producing high temporal resolution images. Taking as input $\mathbf{X}$ or $\hat{\mathbf{X}}$, the forward pass of each branch is identical, so we explain only one of these in the following.

We first employ a frozen image encoder of type CLIP $\mathcal{V}(\cdot)$ [22] to encode $\mathbf{X}$ or $\hat{\mathbf{X}}$ into an embedding space. Subsequently, a learnable MLP is used to project the image embedding into implicit text embeddings. We use $\mathbf{X}$ or $\hat{\mathbf{X}}$ along with the implicit text embeddings as input to a text-to-image diffusion model [17] for feature extraction. More precisely, we employ a UNet architecture to do the denoising process. Formally for the image branch, we have:

$$\mathbf{f} = F_{\boldsymbol{\theta}_U}(\mathbf{X}, \mathrm{MLP}(\mathcal{V}(\mathbf{X}))), \tag{1}$$

where $\mathbf{f}$ is the feature vector from the diffusion network in the image branch, and $\boldsymbol{\theta}_U$ is the parameters of the UNet model. We feed this feature vector $\mathbf{f}$ as input to a Mask2Former model [7] to produce class-agnostic binary masks with their corresponding mask embeddings. For categorization, we use a text encoder of type CLIP $\mathcal{T}(\cdot)$ to embed the categories in $\mathcal{C}^{seen}$. We thus perform a dot product between text and mask embeddings to categorize each mask.

We use the estimated segmentation map of the image branch as the ground truth of the event branch by computing a loss function between both outputs. However, as we kept E2VID frozen, under poor reconstructions, we weighed this loss function using the output of a dissimilarity network to give more emphasis to the regions where E2VID reconstructs well. We further perform knowledge distillation from the Mask2Former in the image to the Mask2Former in the event branch, and similarly for the MLP networks. The parameters of the two branches are initialized with the weights of the ODISE model [34].

3.3 Distilling Image Embeddings

To address the difference in image embeddings between the output of the MLP for the grayscale image and the corresponding output for the synthetic (reconstructed) image, we implement knowledge distillation. This involves transferring knowledge from the image embeddings of the original image to those of the reconstructed image in the event branch. To do so, we introduce the minimization of the Frobenius norm ($\| \cdot \|_F$) of the matrix of differences between real and reconstructed images encoded by the trainable MLP:

$$\mathcal{L}_t = \| \, \mathrm{MLP}_{\mathbf{X}}(\mathcal{V}(\mathbf{X})) - \mathrm{MLP}_{\mathbf{E}}(\mathcal{V}(\hat{\mathbf{X}})) \|_F, \tag{2}$$

where $\mathrm{MLP}_{\mathbf{X}}(\cdot)$ is the frozen MLP of the image branch and $\mathrm{MLP}_{\mathbf{E}}(\cdot)$ is the MLP of the event branch. In other words, we leverage the information encoded in the image embeddings of the original image to guide the learning of the image embeddings for the reconstructed image. The Frobenius norm serves as a metric to quantify the dissimilarity between these embeddings, enabling the model to refine its representation of image information and enhance the consistency between the two image modalities.

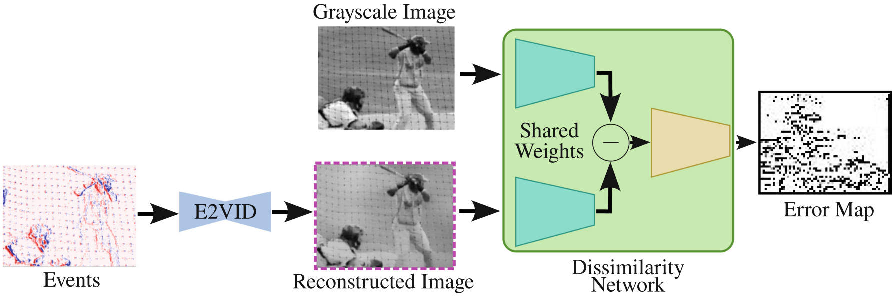


Fig. 3. Dissimilarity network takes the grayscale and reconstructed images as input, and it outputs an error map to reweight the mask loss. E2VID is unable to reconstruct the stripes and hence considered a high error area by the dissimilarity network

3.4 Feature Distillation

To provide further guidance from the image branch to the event one, we minimize the Frobenius norm of the matrix differences of the outputs of each layer in the transformer decoder of Mask2Former between the original image and reconstructed image:

$$\mathcal{L}_f = \frac{1}{L}\sum_{i=1}^{L} \|\mathbf{D}_i - \widehat{\mathbf{D}}_i\|_F\,, \tag{3}$$

where L represents the total number of decoder layers, and $\mathbf{D}$ and $\widehat{\mathbf{D}}$ are the output matrices of each layer in the image and events branch, respectively.

3.5 Mask Re-weighting

While maintaining E2VID frozen, poor reconstructions may occur, prompting the imposition of classification on inadequately reconstructed regions. However, this approach risks compromising model performance where reconstructions are accurate. To address this, we introduce a dissimilarity network to discern differences between the grayscale image and its reconstructed counterpart from events. Illustrated in Fig. 3, this network comprises two convolutional layers. The first layer shares weights for grayscale and reconstructed images, followed by a rectified linear unit (ReLU) activation, while the second layer is followed by a sigmoid activation function $\sigma(\cdot)$. The squared error between the grayscale and reconstructed images' outputs feeds into the second convolutional layer, generating a reweighting map. As shown in Fig. 3 that a notable discrepancy exists between the grayscale image and the reconstructed image, particularly concerning the stripes on the shirt in this example. Consequently, this discrepancy in the error map indicates reduced importance attributed to that specific area. Mathematically, this process can be expressed as:

$$\mathbf{M} = \sigma(\text{conv}_2(\text{ReLU}(\text{conv}_1(\mathbf{X})) - \text{ReLU}(\text{conv}_1(\hat{\mathbf{X}})))^2), \tag{4}$$

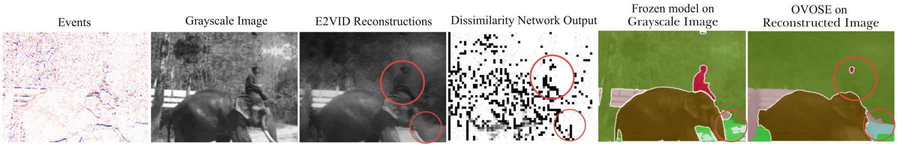

Events
Grayscale Image
E2VID Reconstructions
Dissimilarity Network Output
Frozen model on Grayscale Image
OVOSE on Reconstructed Image

Fig. 4. The impact of reweighting the mask loss, influenced by the dissimilarity between the grayscale and reconstructed images. Poorly reconstructed areas such as the person and the elephant's trunk lead to their exclusion in the reweighting process

where $\mathbf{X}$ is the grayscale original image and $\hat{\mathbf{X}}$ is the reconstructed image from E2VID. We use $\mathbf{M}$ to re-weight our distillation loss at the level of segmentation maps $\mathcal{L}_m$. This loss for the ith stream is given by:

$$\mathcal{L}_m = \mathbf{M} \odot \mathcal{L}_{CE}\left(\mathbf{Y}_i, \hat{\mathbf{Y}}_i\right), \tag{5}$$

where $\mathbf{Y}_i \in \mathbb{R}^{H \times W \times |\mathcal{C}^{seen}|}$ is the output segmentation map of the image branch, $\hat{\mathbf{Y}}_i \in \mathbb{R}^{H \times W \times |\mathcal{C}^{seen}|}$ is the output segmentation map of the event branch, $\mathcal{L}_{CE}$ is the cross-entropy loss, and $\odot$ is the point-wise product between matrices. We illustrate a sample output of the dissimilarity network in Fig. 4. It can be seen that OVOSE successfully ignores the areas where the error is high, for example, the person and the elephant's trunk.

3.6 Category Label Supervision

As we have access to category labels in the training set, we follow [34] to compute a category label loss $\mathcal{L}_c$. To this end, we compute the probability of a mask belonging to one of the training categories using a cross-entropy loss with a learnable temperature parameter as in [34].

The total loss of OVOSE is given by:

$$\mathcal{L}_{final} = \mathcal{L}_t + \mathcal{L}_f + \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c \tag{6}$$

where $\lambda_m = 5.0$ and $\lambda_c = 2.0$ are regularization parameters.

4 Experiments and Results

In this section, we present an overview of the baseline methods for comparison, details on the training and test data, and a comprehensive analysis of both quantitative and qualitative results. Subsequently, we delve into ablation studies to validate the components of OVOSE.

4.1 Experimental Framework

Baseline Methods. As there is no open-vocabulary semantic segmentation method for events, we benchmark OVOSE against leading UDA methods for semantic segmentation in event-based data. However, this comparison is unfair with OVOSE since UDA methods: (i) know the set of unseen classes, (ii) have access to one or multiple labeled source datasets, and (iii) have access to the unlabeled testing dataset for adaptation. We also compare our algorithm with straightforward adaptations of open-vocabulary semantic segmentation methods in regular images. This adaptation consists of reconstructing a grayscale image from the stream of events with the E2VID model (similar to our event branch) and using this as input to the open-vocabulary model. More precisely, OVOSE is compared with the UDA methods E2VID [25], EV-transfer [20], VID2E [11] and ESS [28]. For the open-vocabulary methods, we compare our algorithm against E2VID+OpenSeed [36] and E2VID+ODISE [34]. We use the mean Intersection over Union (mIoU) and pixel accuracy metrics for the quantitative evaluations.

Training Data. As described in the problem statement, OVOSE requires a training dataset $\mathcal{X}_{train}$ where we have the stream of events and their corresponding grayscale images. To address this need, we leverage synthetic training data as introduced by [25]. This synthetic dataset was generated using the event simulator ESIM [24], which simulated MS-COCO images [18]. The dataset comprises $1,000$ sequences, each spanning 2 seconds, with grayscale images and events of dimensions 240×180. For our training purposes, we resize the images and events to dimensions 256×192, ensuring compatibility and optimization for our network. Following [34], we use MS-COCO classes for category label supervision.

Evaluation Datasets. We evaluate the open-vocabulary performance of OVOSE on two popular event camera-based self-driving datasets and Time-Lens++:

- *DAVIS Driving Dataset* (DDD17) is a dataset for semantic segmentation in autonomous driving. Alonso et al. [1] pre-trained an Xception network to generate semantic pseudo-labels which were consolidated into six classes: flat (road and pavement), background (construction and sky), object (pole, pole group, traffic light, traffic sign), vegetation, human, and vehicle.
- *DSEC-Semantic* DSEC [13] consists of high-resolution images 1440×1080, synchronized events 640×480, and semantic labels [28] are generated using a state-of-the-art semantic segmentation method. The fine-grained labels for 19 classes are consolidated into 11 classes: background, building, fence, person, pole, road, sidewalk, vegetation, car, wall, and traffic sign.
- *TimeLens++* We show qualitative results of applying directly OVOSE in the Time lens++ dataset [29]. Time Lens++ consists of high-resolution events of size 970×625.

Implementation Details. We keep the image branch frozen and fine-tune the MLP and Mask2Former of the event branch. We use convolutions with 3×3 kernel size with a stride of 2 and padding 1 for the dissimilarity network. We set the learning rate as 1×10^{-5}. We train OVOSE using Adam optimizer with a batch size of 4 on Nvidia Ampere GPU with 48GB of RAM. We initialize the weights of text and image encoder from [22], text-to-image diffusion UNet from [23], and MLP and Mask2former from [34]. For qualitative results we use the open source code and weights provided by [28,34]. For OpenSeed [36], we use their provided open-source code and weights.

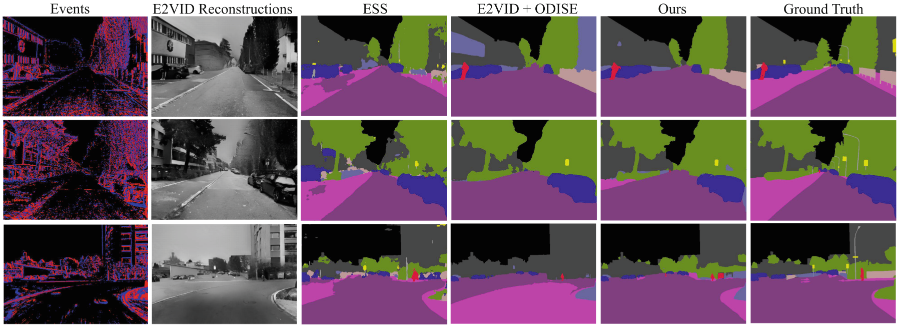


Fig. 5. Qualitative samples from ESS in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. As compared to ESS and E2VID+ODISE, OVOSE produce accurate and less noisy predictions even though it is trained on a synthetic dataset

4.2 Results

Semantic Segmentation on DSEC-Semantic. Figure 5 shows a qualitative comparison of OVOSE against ESS and ODISE. We obtain the qualitative results for the ESS in Fig. 5 by utilizing the official model provided by the authors. Even though the ESS method is trained on real events in a closed-set setting, its overall semantic segmentation across the entire image appears noisy. Notably, it fails to segment vehicles in several instances accurately. For E2VID+ODISE in an open vocabulary setting, it misclassifies parts of the road and buildings. In contrast, OVOSE excels in recognizing traffic signs and delivers superior overall semantic segmentation across the entire image. This is particularly evident in its ability to discern intricate details and provide accurate segmentations, showcasing its robustness and adaptability in the driving scenario of DSEC. Table 1 presents the comparison of OVOSE against the baseline methods on the DSEC-Semantic dataset. Even though the UDA methods are trained in the closed-set setting using real events and urban street datasets similar to DSEC, OVOSE surpasses their performance by a significant margin. Notably, our model improves the state-of-the-art UDA method ESS [28] by a substantial 3.57% in the mIoU metric. Similarly, OVOSE outperforms E2VID+ODISE by 4.83% in the mIoU metric, showcasing the superior performance of OVOSE for semantic segmentation in event-based data.

Semantic Segmentation on DDD17. Figure 6 presents a qualitative comparison between OVOSE, ESS, and E2VID+ODISE. ESS effectively segments poles and other event-specific objects that may not be available in the ground truth. However, it struggles with inaccuracies and noise when segmenting cars. In contrast, OVOSE operating in an open vocabulary setting showcases the ability to segment traffic signs even when unlabeled in the ground truth images. Notably, OVOSE excels in differentiating between classes, particularly with vehicles, and demonstrates more accurate and noise-resistant semantic segmentation compared to the ESS. OVOSE also outperforms E2VID+ODISE by providing

Table 1. Results on DSEC Semantic in UDA and open-vocabulary setting. OVOSE not only outperforms all the UDA methods even though they are trained in closed-set settings but also translated open-vocabulary methods

Method	Type	Problem Formulation	Training Data	Input	Acc (%) ↑	mIoU (%) ↑
EV-Transfer [20]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	60.50	23.20
E2VID [25]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	76.67	40.70
ESS [28]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	84.04	44.87
E2VID+OpenSeed [36]	Translation	Open-Set	MS-COCO	events	65.25	32.82
E2VID+ODISE [34]	Translation	Open-Set	MS-COCO	events	81.24	43.61
OVOSE (Ours)	Distillation	Open-Set	EV-COCO	events	**85.67**	**48.44**

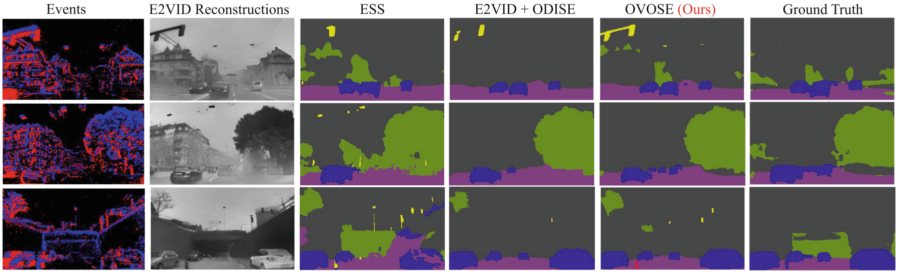


Fig. 6. Qualitative samples from DDD17 in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. OVOSE does better overall predictions, especially vehicles, persons, vegetation, and construction

clearer object recognition and segmentation. This comparison emphasizes the need for knowledge distillation and OVOSE's capacity to deliver precise semantic segmentation even in scenarios with incomplete or noisy ground truth annotations.

When it comes to quantitative evaluation on the DDD17 dataset, the presence of noisy ground truth labels, as highlighted by [28], poses challenges. Additionally, the previous work by Ev-SegNet [1] merged several labels, further complicating the evaluation process. To address this, we consider the original classes pre-merge for predictions, subsequently merging them to facilitate a comparison with the ground truth. Table 2 summarizes the quantitative comparison of OVOSE with the baseline methods on the DDD17 dataset. OVOSE significantly improves the UDA state-of-the-art ESS method [28] by 0.93% in the mIoU metric. Notably, our algorithm outperforms E2VID+OpenSeed and E2VID+ODISE by large margins, underscoring the significance of knowledge distillation and mask reweighting in semantic segmentation for event cameras.

Qualitative Results on the Time Lens++ Dataset. We use the Time Lens++ [29] dataset to evaluate the open-vocabulary capabilities of OVOSE since no open-vocabulary event dataset is available. Some qualitative results are illustrated in Fig. 7. We see in Fig. 7 that OVOSE performs high-quality segmentation of various classes such as trees, traffic lights, people, cars, buildings, sidewalk, and roads which demonstrates OVOSE preserved open vocabulary

Table 2. Results on DDD17 in UDA and open-vocabulary setting. OVOSE outperforms all the UDA methods trained in closed-set settings but also translated open vocabulary methods

Method	Type	Problem Formulation	Training Data	Input	Acc (%) ↑	mIoU (%) ↑
EV-Transfer [20]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	47.37	14.91
E2VID [25]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	83.24	44.77
VID2E [11]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	85.93	45.48
ESS [28]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	87.86	52.46
E2VID+OpenSeed [36]	Translation	Open-Set	MS-COCO	events	33.25	17.95
E2VID+ODISE [34]	Translation	Open-Set	MS-COCO	events	84.63	48.12
OVOSE (Ours)	Distillation	Open-Set	EV-COCO	events	**88.84**	**53.39**

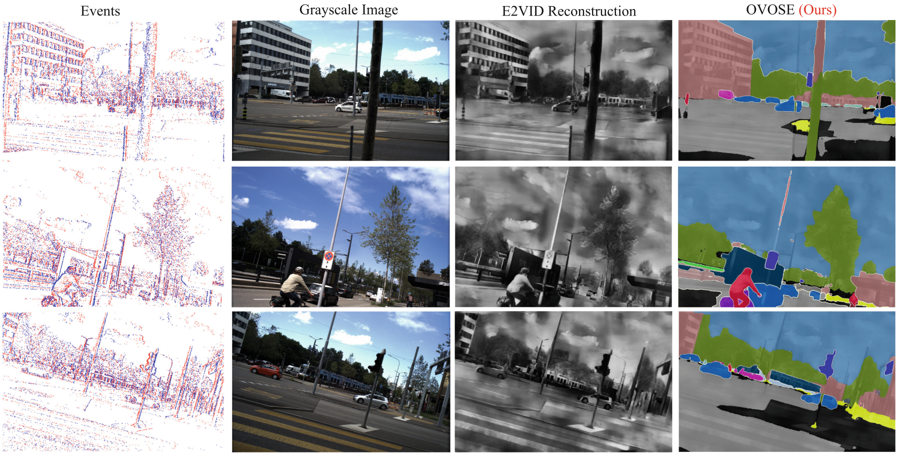


Fig. 7. Open-vocabulary performance of OVOSE on the Time Lens++ dataset [29]

capabilities. Furthermore, OVOSE can recognize the bike and train even though the reconstruction from E2VID is very noisy and far from optimal.

4.3 Ablation Studies

Mask Reweighting. We conduct an ablation study to validate the reweighting schemes and knowledge distillation techniques in OVOSE. To that end, We evaluate OVOSE without training; this is equivalent to performing E2VID+ODISE. Then, we analyzed the impact of reweighting by using only the distillation pipeline, with reweighting based upon cosine similarity (CS) on the original and reconstructed image, by using the squared differences of the output of Stable Diffusion (SD) from the image and event branches and finally with our Dissimilarity Network (DN) corresponding to OVOSE. Table 3 shows the results of the ablation study. Distillation resulted in a notable increase in accuracy by 3.8% and mIoU by 3.4%, showcasing the effectiveness of this approach. Moreover, the introduction of the dissimilarity network for reweighting the loss function gives an additional improvement of 1.43% in the mIoU metric.

Table 3. Ablation of distillation strategies of OVOSE on DSEC Semantic dataset

Method	Acc (%) ↑	mIoU (%) ↑
Baseline	81.2	43.61
Distillation	85.0	47.01
Distillation+Reweight CS	85.1	48.01
Distillation+Reweight SD	85.1	47.32
Distillation+Reweight DN	**85.6**	**48.44**

Table 4. Ablation studies on DSEC-Semantic dataset. MLP's is MLP layer of OVOSE and MG is the Mask Generator

Ablation	Parameters		Acc (%) ↑	mIoU (%) ↑
Loss	$\boldsymbol{\lambda_c}$	$\boldsymbol{\lambda_m}$		
	5.0	5.0	84.47	46.47
	5.0	2.0	84.70	46.44
	2.0	5.0	**85.67**	**48.44**
Finetuning	**MLP's**	**MG**		
	✓	✗	82.46	44.34
	✗	✓	83.19	44.68
	✓	✓	**85.67**	**48.44**
Image-Reconstructor	**FireNet** [27]	**E2VID** [25]		
	✓	✗	82.85	43.94
	✗	✓	**85.67**	**48.44**
Text Prompts	**DSEC Classes**	**Ours**		
	✓	✗	84.04	46.50
	✗	✓	**85.67**	**48.44**

Loss Parameters. We study the impact of varying λ_c and λ_m in Eq. 6 on model performance. As shown in Table 4, the optimal combination of $\lambda_c = 2.0$ and $\lambda_m = 5.0$ achieved the highest accuracy 85.67% and mean Intersection over Union (mIoU) 48.44%, indicating that a lower weight on caption loss and a higher weight on mask loss enhance performance.

Fine-Tuning Ablation. We explore the effects of fine-tuning the MLP layers and the Mask Generator (MG), individually and together. From Table 4, we observe that fine-tuning only the MLP resulted in 82.46% accuracy and 44.34% mIoU, while fine-tuning only MG slightly improved performance to 83.19% accuracy and 44.68% mIoU. However, simultaneous fine-tuning of both MLP's and MG yielded the best results with 85.67% accuracy and 48.44% mIoU, underscoring the necessity of fine-tuning both components together.

Image Reconstructor Ablation. We replaced E2VID [25] with FireNet [27] and reported results in Table 4. E2VID significantly outperformed FireNet, achieving 85.67% accuracy and 48.44% mIoU which indicates that E2VID's recurrent neural network handles temporal information well and outputs higher quality reconstructions for our downstream task.

Ablation on Text prompts. We further assess the effectiveness of two text prompt configurations: directly using the DSEC-Class name and our text prompts shown in Tables 1 and 2 in the supplementary document. Results in Table 4 show that DSEC-Classes configuration resulted in 84.04% accuracy and 46.50% mIoU. In contrast, our text prompts configuration achieved superior performance with 85.67% accuracy and 48.44% mIoU, demonstrating that the detailed prompts enhance the model's performance more effectively.

5 Conclusion

In this work, we introduced OVOSE, the first open-vocabulary semantic segmentation algorithm designed for event-based data. Comprising grayscale image and event branches, each equipped with a pre-trained foundation model, our approach leverages synthetic data for Knowledge Distillation from regular images to enhance semantic segmentation in events. OVOSE employs distillation at multiple stages of the foundation model, enhancing its effectiveness with a mask reweighting strategy through a dissimilarity network. We evaluate OVOSE in the DDD17 and DSEC-Semantic datasets and compare it against with existing methods in UDA close-set semantic segmentation and foundation models adapted to the event domain with E2VID. Our algorithm outperforms all these models, offering a promising avenue for research in open-vocabulary semantic segmentation tailored for event cameras.

Acknowledgments. We acknowledge financial support from the PNRR MUR project PE0000013-FAIR and from the Sapienza grant RG123188B3EF6A80 (CENTS). We acknowledge the support by the French National Research Agency (ANR) with the ANR-20-CE23-0027. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

References

1. Alonso, I., Murillo, A.C.: EV-SegNet: semantic segmentation for event-based cameras. In: IEEE/CVF CVPRW (2019)
2. Binas, J., Neil, D., Liu, S.C., Delbrück, T.: Ddd17: End-to-End Davis Driving Dataset (2017). ArXiv : arxiv.org/abs/1711.01458
3. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. Adv. Neural Inf. Process. Syst. **32**, (2019)
4. Chen, G., Cao, H., Conradt, J., Tang, H., Rohrbein, F., Knoll, A.: Event-based neuromorphic vision for autonomous driving: a paradigm shift for bio-inspired visual sensing and perception. IEEE Signal Process. Mag. **37**(4), 34–49 (2020)

5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision ECCV 2018, p. 833–851. Springer-Verlag, Berlin, Heidelberg (2018)
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 CVPR, pp. 1290–1299 (2022)
8. Cho, H., Kim, H., Chae, Y., Yoon, K.J.: Label-free event-based object recognition via joint learning with image reconstruction from events. In: Proceedings of the IEEE/CVF ICCV, pp. 19866–19877 (2023)
9. Dosovitskiy, A., et al.: An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale (2020). arXiv preprint: arXiv:2010.11929
10. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., et al.: Event-based vision: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 154–180 (2020)
11. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to events: recycling video datasets for event cameras. In: Proceedings of the IEEE/CVF CVPR, pp. 3586–3595 (2020)
12. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings of the IEEE/CVF ICCV, pp. 5633–5643 (2019)
13. Gehrig, M., Aarents, W., Gehrig, D., Scaramuzza, D.: Dsec: a stereo event camera dataset for driving scenarios. IEEE Robot. Autom. Lett. **6**, 4947–4954 (2021)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network (2015). arXiv preprint: arXiv:1503.02531
15. Jian, D., Rostami, M.: Unsupervised domain adaptation for training event-based networks using contrastive learning and uncorrelated conditioning. In: Proceedings of the IEEE/CVF ICCV, pp. 18721–18731 (2023)
16. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: ICLR (2022)
17. Liang, F., et al.: Open-vocabulary semantic segmentation with mask-adapted clip. In: 2023 IEEE/CVF CVPR, pp. 7061–7070 (2022)
18. 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: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, pp. 740–755. Springer (2014)
19. Messikommer, N., et al.: Multi-bracket high dynamic range imaging with event cameras. In: 2022 IEEE/CVF CVPRW, pp. 546–556 (2022)
20. Messikommer, N., Gehrig, D., Gehrig, M., Scaramuzza, D.: Bridging the gap between events and frames through unsupervised domain adaptation. IEEE Robot. Autom. Lett. **7**(2), 3515–3522 (2022)
21. Park, S., Kwak, N.: Feed: Feature-Level Ensemble for Knowledge Distillation (2019). arXiv preprint: arXiv:1909.10754
22. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML, pp. 8748–8763. PMLR (2021)
23. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical Text-Conditional Image Generation with Clip Latents, vol. 1, no. 2, pp. 3 (2022). arXiv preprint arXiv:2204.06125

24. Rebecq, H., Gehrig, D., Scaramuzza, D.: Esim: an open event camera simulator. In: Conference on Robot Learning (2018)
25. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (2019)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF CVPR, pp. 10684–10695 (2022)
27. Scheerlinck, C., Rebecq, H., Gehrig, D., Barnes, N., Mahony, R., Scaramuzza, D.: Fast image reconstruction with an event camera. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 156–163 (2020)
28. Sun, Z., Messikommer, N., Gehrig, D., Scaramuzza, D.: Ess: Learning event-based semantic segmentation from still images. In: ECCV (2022)
29. Tulyakov, S., Bochicchio, A., Gehrig, D., Georgoulis, S., Li, Y., Scaramuzza, D.: Time Lens++: event-based frame interpolation with non-linear parametric flow and multi-scale fusion. In: IEEE/CVF CVPR (2022)
30. Wang, L., Chae, Y., Yoon, K.J.: Dual transfer learning for event-based end-task prediction via pluggable event to image translation. In: Proceedings of the IEEE/CVF ICCV, pp. 2135–2145 (2021)
31. Wang, L., Chae, Y., Yoon, S.H., Kim, T.K., Yoon, K.J.: Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 608–619 (2021)
32. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: Proceedings of the IEEE/CVF CVPR, pp. 8256–8265 (2019)
33. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge distillation meets self-supervision. In: European Conference on Computer Vision, pp. 588–604. Springer (2020)
34. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF CVPR, pp. 2955–2966 (2023)
35. Yang, Y., Pan, L., Liu, L.: Event camera data pre-training. In: Proceedings of the IEEE/CVF ICCV, pp. 10699–10709 (2023)
36. Zhang, H., Li, F., Zou, X., Liu, S., Li, C., Yang, J., Zhang, L.: A simple framework for open-vocabulary segmentation and detection. In: Proceedings of the IEEE/CVF ICCV, pp. 1020–1031 (2023)
37. Zhao, L., Peng, X., Chen, Y., Kapadia, M., Metaxas, D.N.: Knowledge as priors: cross-modal knowledge generalization for datasets without superior knowledge. In: 2020 IEEE/CVF CVPR, pp. 6527–6536 (2020)
38. Zhu, A.Z., Yuan, L., Chaney, K., Daniilidis, K.: Unsupervised event-based learning of optical flow, depth, and egomotion. In: Proceedings of the IEEE/CVF CVPR, pp. 989–997 (2019)

Moiré Pattern Detection: Stability and Efficiency with Evaluated Loss Function

Zhuocheng Li[1], Xizhu Shen[1], Simin Luan[1], Shuwei Guo[1], Zeyd Boukhers[2,3],
Wei Sui[1], Yuyi Wang[4], and Cong Yang[1](✉)

[1] Ecology and Innovation Center of Intelligent Driving (BeeLab), Soochow University, Suzhou, China
{2262404048,2262401045}@stu.suda.edu.cn, cong.yang@suda.edu.cn
[2] Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany
zeyd.boukhers@fit.fraunhofer.de
[3] University Hospital of Cologne, Cologne, Germany
[4] CRRC Zhuzhou Institute Co., Ltd., Zhuzhou 412001, Hunan, China

Abstract. Detecting moiré patterns in digital images is essential as it offers insights for assessing image integrity and undertaking demoiréing processes. MoireDet is a simple and efficient moiré pattern detection neural network designed explicitly for moiré edge map estimation [23]. However, the random feature mapping of the Performer significantly affects the prediction for continuous video frames. This paper introduces MoireDet+ based on existing work, which introduces Vision Transformer into moiré-related tasks. MoireDet+ utilizes a mixed-encoder as a backbone, integrating both high- and low-level vision encoders in an FPN-like method, along with a spatial encoder to extract the complex spatial features of moiré patterns. Furthermore, we produce a rapid approximation-based evaluation method to aid loss function design in image restoration and similar tasks. MoireDet+ reaches a state-of-the-art level on mainstream datasets, reducing time costs by 18% compared to MoireDet and other baselines with comparable performance levels.

Keywords: Moiré Pattern · Moiré Pattern Detection · Moiré Pattern Restoration

1 Introduction

Demoiréing has been a long-standing research direction in image restoration research. Moiré patterns arise from the loss of texture detail due to the Nyquist effect [2]. It is caused by an insufficient sampling frequency of the camera sensor, which occurs when photographing objects with high-frequency appearance features, such as displays and high-frequency fine lines in nature. The moiré pattern detection and demoiré tasks target the same subject matter but have different

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 34–49, 2025.
https://doi.org/10.1007/978-3-031-78444-6_3

objectives. Moiré pattern detection involves binary classification to detect the presence of moiré patterns in images. This study does not encompass the widely researched field of demoiré but focuses solely on moiré pattern detection. Moiré pattern detection task has applications in diverse fields [1,11,23], including portal security, autonomous driving, etc. Typical work scenarios include (1) Face moiré detection for cameras. Here, moiré patterns can be an anti-cheating measure against attempts to spoof the camera using images of the owner's face [11]. (2) Moiré extraction in fashion design. When designers show particular styles by designing moiré, the moiré pattern can help designers evaluate and refine their complex designs. (3) Remove misleading information from autopilot tasks. In bustling commercial centers, images such as cars displayed by giant light signs on the roadside can deceive multi-camera systems, leading to errors in autopilot functionality.

MoireDet by Yang et al. [23] has demonstrated efficiency in detecting moiré patterns and identifying moiré images. By leveraging both high-level contextual features and low-level structural features, MoireDet employs three encoders to encode moiré patterns effectively, utilizing adaptive kernels that are sample- and location-specific. However, providing a high-quality prediction of moiré pattern for a screen-shot video can be challenging for MoireDet. The random feature mapping of the Performer significantly affects predictions for continuous video frames, leading predicted videos to exhibit a ”jittering” effect of moiré. This problem becomes more pronounced when MoireDet is compared to other models designed for similar purposes, emphasizing the complexity of accurately detecting video moiré patterns. The prediction of the moiré pattern is often not fine enough compared to the ground truth, underscoring the need for further refinement in model performance.

Succinctly, the main contributions of this work are as follows: (1) we introduce a new type of Vision Transformer inspired by MB-TaylorFormer [22], a proven effective image denoising method for finely extracting fusion features. (2) We enhance the structure of the original model by using a Mixed-Encoder as the backbone, which integrates both high- and low-level vision encoders. This is achieved by using an FPN-like method while keeping a spatial encoder to extract the complex spatial features of moiré patterns. In addition, a distinctive contrast detection module during the prediction stage is proposed in MoireDet+. We show superior performances against the state-of-the-art MoireDet on moiré pattern detection and identification (e.g., SSIM 0.57 v.s. 0.33 on MoireScape dataset, accuracy 82.1% VS 81.6% on moireIDT dataset), a prominent improvement in accuracy, and the time cost is 18% lower than MoireDet on prediction period. The visualization of MoireDet+ output can be seen in Fig. 1.

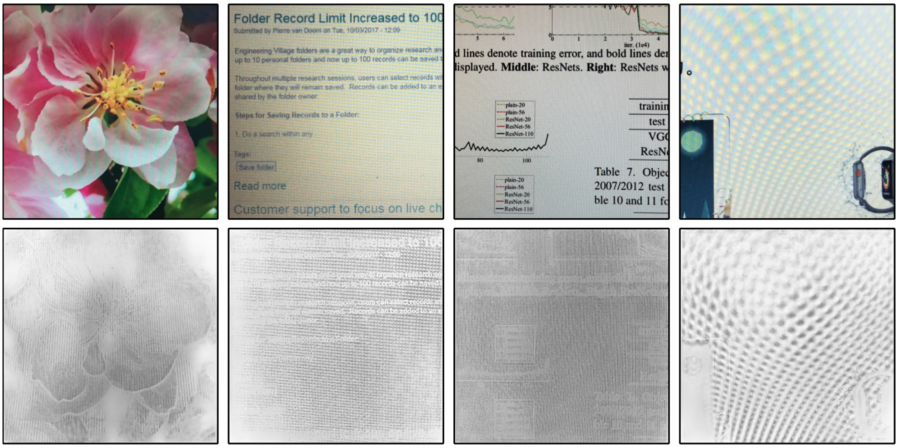

Folder Record Limit Increased to 100
Read more
Customer support to focus on live ch
d lines denote training error, and bold lines de
isplayed. Middle: ResNets. Right: ResNets w
Table 7. Objec
2007/2012 test
ble 10 and 11 fo

Fig. 1. Examples of moiré patterns detected by MoireDet+. **Top:** Model's input, and **Bottom:** Corresponding Moiré Layers by MoireDet+)

2 Related Works

2.1 Moiré Pattern Detection

While considerable research has been devoted to removing moiré [10,21,24], moiré detection methods have received comparatively less attention. Historically, moiré detection methods were generally extended from existing detection methods like smoke, haze, etc. [1,18]. Later studies explored multi-scale detection methods using the U-Net class, which are significantly effective and can be migrated to moiré detection using the shadow detection method through RNN [14]. Abraham [1] and Garcia [11] proposed an intriguing approach combining neural networks with spectral analysis to portray the full range of moiré features. But MoireDet [23] employs predicted grey-scale moiré images for detection, confirming that the direct moiré detection method is ineffective in moiré detection tasks, while the prediction-then-detection method fits better. Our proposed MoireDet+ is built on a single Vision Encoder and Vision Transformer to address sensitivity issues. MoireDet+ has a state-of-the-art performance on primary datasets.

2.2 Loss Function Set

In previous research, the setting of the loss function has often been standardized within specific tasks [6,9]. However, image generation tasks require different loss functions tailored to their unique requirements [20]. While standard loss like SSIM offers reasonable outcomes, relying solely on SSIM can lead to issues such as learning overly complex features and suppressing model output [17]. It is crucial to adopt an appropriate loss function to guide the generation of outputs that align with the specific features of the task. Still, this process may yield a less-than-ideal loss function, possibly leading to misleading results or insufficient smoothing of the gradient descent. Consequently, establishing a method to evaluate the loss function is essential for task-specific settings. We use mathematical analysis to derive and validate this method, as detailed in Sect. 4.

3 MoireDet+

Our proposed MoireDet+ employs distinct methods for the output moiré **O** and detection moiré stages (see Fig. 2). In the output stage, we utilize Mixed-feature Encoder and Spatial Encoder to extract moiré features and generate a binary output map. In the detection stage, we conduct comparative moiré detection on **O** through the tail-plugging detection module. Also, the detection method of the tail-plugging module can be changed according to specific usage scenarios.

3.1 Architecture

Mixed-Encoder. The Mixed-Encoder(ME) in MoireDet+ is a branch encoder that encodes fine multi-scale texture details within an image. ME uses the first two residual blocks of ResNet18 [13] and the complete ResNet18 to extract both low-level and high-level features from the original image. These features are then fused using BiFPN [4], the architecture of ME is depicted in Fig. 3. Unlike MoireDet, which fuses both high- and low-level encoders (HLE and LLE) via pointwise multiplication, a method that can lead to complications during gradient descent optimization, we enhance this process with a continuous serial fusion strategy. This refinement improves parameter-to-loss gradient mixing, strengthening the global optimization process following feature map fusion.

Vision Transformer. We introduce a new type of Vision Transformer inspired by MB-TaylorFormer [22], proven effectiveness in image denoising and fine extraction of fusion features. Leveraging Taylor expansion, this approach significantly reduces the variance of similar image features and provides more stable outputs, making MoireDet+ the first model to provide stable restoration for video moiré. Using MB-TaylorFormer, we compress information within feature maps, producing detailed fused feature maps encompassing local texture and macro features. Also, it reduces the time complexity of Vision Transformer's softmax stage from $O(n^2)$ to $O(n)$ [22], substantially decreasing the time cost for MoireDet+ by up to 18% in the same prediction task (see Table 1).

Spatial-Encoder. The Spatial-Encoder (SE) in MoireDet+ is similar to MoireDet and is tasked with indicating the spatial distribution of moiré patterns. Comprising adaptive 5×5 convolution kernels, SE calculates the weights of these kernels based on the output of the Mixed-Encoder feature map. Each convolutional kernel in the SE is responsible for extracting features from specific local regions of the image, enabling the capture of local moiré patterns. In addition to the original input, it also needs to receive the feature map output from the ME for guidance. The spatial pattern of Moiré can be seen in the first row of Fig. 5.

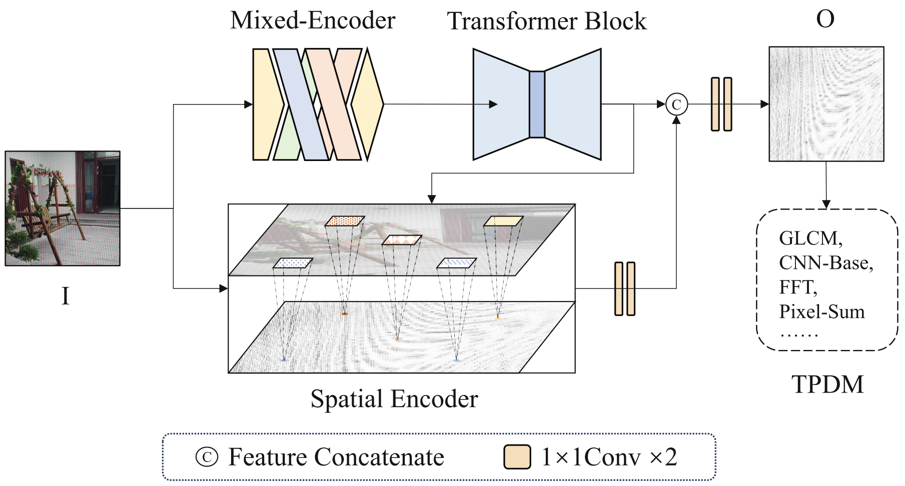


Fig. 2. Architecture of our proposed MoireDet+

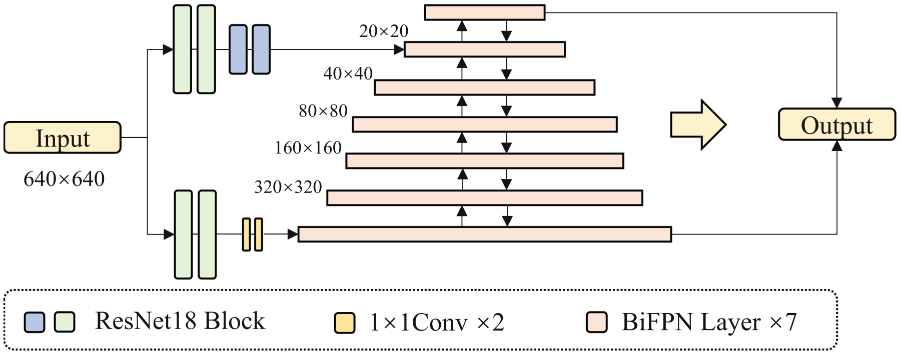


Fig. 3. Mixed-Encoder Architecture

Tail Plug Detection Module. The Tail Plug Detection Module (TPDM) is the primary difference between MoireDet+ and MoireDet in their detection methods. Its purpose is to detect the presence of moiré patterns in the original image based on the extracted moiré layer. This step is necessary because high-frequency textures in the background of the original image can sometimes be mistaken for moiré patterns and extracted incorrectly. Using the moiré layer without verification can lead to false positives. For MoireDet+, we have designed various TPDM methods, including previously used Pixel-Sum and FFT methods, as well as our proposed GLCM (Gray-level Co-occurrence Matrix [19]) and CNN (Convolutional Neural Network) methods. The GLCM method extracts Haralick features [12] from the moiré layer's GLCM and uses SVM (Support Vector Machine) [3] for subsequent binary classification. The CNN method employs a simple shallow CNN for classification.

3.2 Loss Functions

The total loss L is defined as a combination of two components:

$$L = w_1 L_{pixel} + w_2 L_{BS} \quad . \tag{1}$$

where L_{pixel} and L_{BS} denote the Per Pixel Loss and Background Similarity Loss, respectively. The weights w_1 and w_2 are empirically set to 0.7 and 0.3, respectively. L_{BS} is designed to mitigate the impact of high-frequency textures in the background image, which are similar to moiré pattern. The parameters w_1 and w_2 are used to balance L_{pixel} and L_{BS}, and they are determined based on the characteristics of the MoireScape dataset. For datasets with stronger moiré patterns or more high-frequency background textures, these parameters need to be adjusted accordingly.

Per Pixel Loss: Per Pixel Loss (PPL) is the fundamental loss function, i.e., a pixel-by-pixel comparison between the output map and the Ground Truth. In our approach, we utilize the L1 Paradigm for smoothing. The formula for PPL is given by:

$$L_{pixel} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} d(O(i,j), M(i,j)) \quad . \tag{2}$$

where m and n are the height and width of the input image. $d(\cdot)$ denotes the L1 Paradigm. $\mathbf{O}$ represents the output moiré layer, and $\mathbf{M}$ denotes the Ground Truth moiré layer in training triplet.

Background Similarity Loss. Background Similarity Loss (BSL) aims to mitigate mispredictions in the image $\mathbf{I}$, featuring high-frequency regions, such as dense foliage or intricate flower details, which may obscure the moiré layer. Leveraging the training triplet (combined image, moiré layer, natural image) from MoireScape, we compute BSL using the natural image after the edge detection process. The formula of BSL is as follows:

$$L_{BS} = -\frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} d(O(i,j), B_{edge}(i,j)) \quad . \tag{3}$$

where B_{edge} represents the natural image with edge detection applied.

4 Loss Function Evaluation

Custom loss functions are commonly employed in image generation and restoration tasks to guide models, enhancing generalization and learning speed. However, this could potentially lead to unforeseen issues. When using MoireDet for inference, it consistently applies a fixed, grid-like pattern across the image's underlying layer, regardless of the type of moiré pattern present. The authors also observed this problem and suggested that the Performer could resolve it. However, this was actually due to the random mapping by the Performer mitigating the issue rather than fundamentally solving it.

It is crucial to emphasize that the motivation behind proposing a mechanism for evaluating loss functions is to address the challenge of repetitive training and conditioning models, particularly considering the significant time and resource costs associated with larger models. Our proposed evaluation method is intended to be lightweight, facilitating the testing to determine whether a singular or combination of loss functions yields specific patterns of locally optimal solutions. Our evaluation method employs a learnable Tensor for rapid approximation. utilizing a Tensor sheet of the same size as the input to simulate the parameter gradient descent process during learning (see Algorithm 1).

Algorithm 1: Loss Evaluation

```
Input:
    Weight Matrix and Output, Target;
    Ground_Truth, GT;
    Loss Function to be evaluated, L;
Output:
    Visualization of Output Map, O
for step $i$ in range(steps) do
   Loss ← L($Target_i$, $GT$)
   Train and Optimize, optimize on $Target_i$
   $Target_{i+1}$ ← $Target_i$ with Optimization
end for
```

The process enables the selection of the composition of the Target Tensor, which consists of all 0, random values and uses image inputs from the corresponding dataset. Given that the potential effects of locally optimal solutions may not always be quantified through data alone, it is essential to visually observe the ultimate impact of the Target and assess SSIM(Structural SMIilarity) concerning the GT(Ground Truth). The visualization of our proposed loss function evaluation is shown in Fig. 4.

4.1 Validity of Loss Function Evaluation

To substantiate the validity of our method, we employ mathematical analysis to establish error limits between our approximate estimation method and the actual value. The detailed proof is presented below:

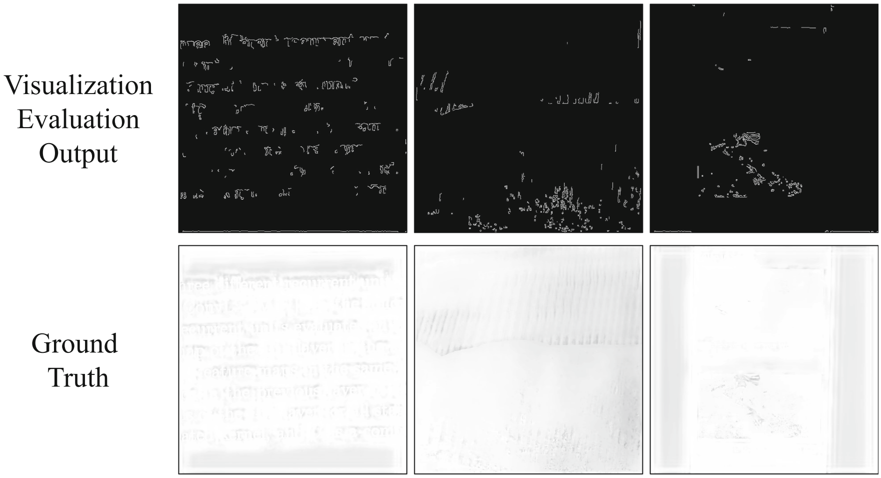


Fig. 4. Visualization of our proposed Loss Function Evaluation. They are displayed on a black background with a pure white input for optimal contrast and clarity

During Training: For a weight parameter w_*:

$$w_{*;k+1} = w_{*;k} \oplus \frac{dL}{dw_{*;k}} \times LearningStep \quad . \tag{4}$$

where $\oplus$ symbolizes the parameter change induced by the Trainer (e.g., ADAM). w denotes a weight parameter in the Learnable Tensor, and k represents the iteration time. L refers to the Loss Function.

During Inference: For a pixel $p_{i,j;k+1}$ in O_{k+1} with the same input I_{Spec}:

$$p_{i,j;k+1} = Net(I_{Spec}; W_{[\cdot N],k}) \quad . \tag{5}$$

where i, j denote the position of pixel p in the map. $W_{[\cdot N],k}$ encompasses all parameters set at iteration k ,ranging from $w_{1;k}$ to $w_{N;k}$. $Net(*;*)$ represents the network used for approximation. I_{Spec} denotes the specific input.This is an abstract reasoning process, which will be approximated in the following steps.

After Approximation: For a pixel $p_{i,j;k+1}$ in O_{k+1} with same input I_{Spec}:

$$p_{i,j;k+1} = p_{i,j;k} - \frac{dL}{dp_{i,j;k}} \quad . \tag{6}$$

To demonstrate that the two methods yield insignificant differences, we compare their outputs, such that:

$$\varDelta p_{i,j;k+1} = \| \; Net(I_{i,j}; W_{[\cdots N];k+1}) - p_{i,j;k} + \frac{dL}{dp_{i,j;k}} \; \|$$

$$= \parallel Net(I_{i,j}; W_{[\cdots N];k}) \oplus \frac{dL}{dW_{[\cdots N];k}} \times \epsilon + \frac{dL}{dp_{i,j;k}} - p_{i,j;k} \parallel \quad . \tag{7}$$

where ϵ represents the learning rate or the change rate of parameter set W after iteration k by the Trainer. $Net(\cdot)$ must exhibit a continuous property, and it is commonly subjected to Lipschitz conditions, which are pivotal in convex optimization. For a network $Net(\cdot)$ that satisfies Lipschitz continuity for all parameters $W_{[...N]}$, it can be derived from the Lipschitz continuity condition that the upper and lower bounds of its variations can be determined:

$$\forall w_* : Net(I_{i,j}; W_{[\cdots N]} \backslash w_{*;k}, w_{k+1}) \leq Net(I_{Spec}; W_{[\cdots N],k}) + L \parallel \Delta w_* \parallel \tag{8}$$

$$\forall w_* : Net(I_{i,j}; W_{[\cdots N]} \backslash w_{*;k}, w_{k+1}) \geq Net(I_{Spec}; W_{[\cdots N],k}) - L \parallel \Delta w_* \parallel \tag{9}$$

where $W_{[\cdots N]} \backslash w_{*;k}, w_{k+1}$ signifies the removal of $w_{*;k}$ and the addition of w_{k+1} in the parameter set $W_{[...N]}$. We also need to ensure that this domain remains sufficiently small:

$$Net(I_{i,j}; w_{[\cdots N];k}) \oplus \frac{dL}{dW_{[\cdots N];k}} \times \epsilon \sim Net(I_{Spec}; w_{[\cdots N],k}) \cdots L \parallel \Delta w_* \parallel$$

$$\Delta w_* = \frac{dL}{dW_*} \times \epsilon \quad . \tag{10}$$

The step length is sufficiently small, Δw_* defines a tight neighborhood size:

$$p_{i,j;k+1} \leq L \parallel \Delta w_{[\cdots N]} \parallel + \frac{dL}{dp_{i,j;k}} \quad . \tag{11}$$

Differences converge rapidly through stepwise training, and adopting a randomized optimization strategy ensures that the output graphs are evenly tuned. It requires the gradient of the Loss Function to be bounded and continuous in our approach to ensure that the range of differences remains reasonably estimable. Failure to satisfy this condition would render the approximation method ineffective. However, it is widely accepted in routine tasks that the gradient of Loss Function is bounded. Hence, in most cases, this condition will be met.

5 Experiments

5.1 Datasets

Moiré Image Identification. Moiré identification task aims to classifier whether an image contains moiré patterns. For this, three datasets (MoireFace [11], MoireIDT [23], and MRBI [25]) are employed for evaluation and comparison. Specifically, the MoireFace dataset was collected using 50 selected original faces displayed on MacBook, iPad, and iPhone for camera capturing with 12 different cameras, display, and distance combinations. 12 × 50 = 600 face images with moiré pattern (positives) were collected. Along with 50 original

Table 1. Ablation Study on MoireDet+, left two columns shows results on MoireScape dataset, right two shows results on Continuous Video Frames

	PSNR	SSIM	RunTime	CVR
MoireDet	8.65	0.33	160.9	0.053
ME+SE	8.84	0.33	122.7	0.055
ME+ViT	10.38	0.37	131.5	**0.032**
MoireDet+	**10.80**	**0.57**	141.3	0.040

faces (negatives), MoireFace contains 600+50 = 650 face images for face-spoofing identification. The MoireIDT dataset comprises 4,000 images collected from various scenarios, including 2000 authentic moiré images (positives) and 2000 moiré-free images (negatives) featuring varying degrees of background complexity. The positive set includes camera-captured screen images and natural images exhibiting moiré effects. Similarly, the MRBI dataset contains 340 pairs for testing.

Moiré Pattern Detection. Moiré pattern detection task aims to detect the distribution and density of moiré patterns (similar to moiré pattern heatmap) within an image. We selected 1000 triplets from the MoireScape dataset [23], using the moiré layer compared with the output to calculate the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural SMIilarity) metrics. In MoireScape, images were organized in triplet form. Each triplet contains a segmentation moiré edge layer with transform, a ground truth background image, and a combined image with the background image and moiré layer. The pure moiré layers were captured from screenshot photos with an all-white screen, followed by edge detection segmentation to isolate the moiré layer. Background images are sourced from COCO [16], ImageNet [7], PASCAL VOC [8] and Retail50K [5] dataset, each resized to 320×320 size. The combined images are generated by transforming the moiré layer with rotation and scaling, reflecting that moiré patterns may not always mask the entire image but only a portion of it.

5.2 Moiré Pattern Detection

First, we train MoireDet+ for high-precision moiré capture. We employ the complete MoireScape dataset [23] with 50000 triplets for training. The training set is divided with a 9 : 1 ratio for training and validation purposes. We use a single NVIDIA RTX 3090 GPU, training lasting 32 h over 100 epochs. The learning rate is set to $1e{-}3$, and the chosen optimizer is Adam [15].

We conducted ablation studies to validate the effectiveness of the MoireDet+ structure. With the complete ME + ViT + SE architecture, the output shows the finest spatial structure and seems most detailed. The model output with Performer is more coarse, while the appearance generally remains like a moiré pattern. Its prediction time cost is about 18%. The result of the ablation study is shown in Table 1.

SE is motivated by the difference in fine-grained moiré patterns by region. During this experiment, we also found that it plays a role in keeping the multi-layer moiré pattern. Multi-layer moiré means that moiré shows different inclinations and appearances but in the same region. With SE, the model can precisely capture the moiré pattern with various patterns, as shown in Fig. 5 (middle). In particular, the captured patterns show a finer appearance. Otherwise, the output is misled by dark regions from the background (see Fig. 5 (right)).

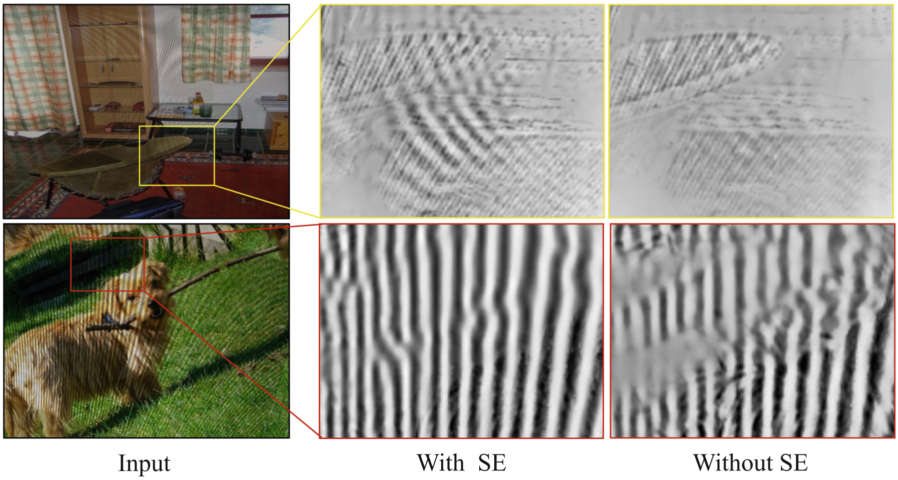


Fig. 5. Example on model with and without SE, the first row shows different inclination, the second row shows SE's finer appearance

Compared to MoireDet, the detected moiré patterns from MoireDet are more fine-grained. Ideally, MoireDet+ reproduces coarse- and fine-grained moiré spatial structures with near pixel-level accuracy, whereas MoireDet only captures rough distribution lines. Adjustments to the loss function effectively resolve periodic lattice point issues seen in MoireDet. Existing models still struggle with occlusion and misrepresentation by dark backgrounds, and MoireDet+ offers more accurate predictions under dark occlusion due to the heightened sensitivity of its Vision Transformer.

5.3 Moiré Image Identification

With the detection output from MoireDet+, we extend it for the moiré image identification task with our proposed TPDM (see Fig. 2). With the distribution and density of detected moire patterns from MoireDet+, we directly identify the existence of moire patterns with TPDM. Our experiments in this part show that in environments with dark and variable background images, MoireDet+ (equipped with TPDM) performs more stability than the other methods.

Table 2. Ablation study of MoireDet+ on moire image identification task with Pixel-Sum method [23] employed

	MoireDet+	ME+ViT	ME+SE
MoireIDT	81.75	79.21	**81.99**
MRBI	**90.00**	85.41	88.19

Table 3. Comparison with four TPDM identification methods on MoireIDT [23] and MRBI [11] dataset

	MoireIDT	MRBI
Pixel-Sum	81.75	90.00
GLCM	80.00	91.52
FFT	81.89	89.77
CNN	**82.10**	**90.42**

Firstly, we do an ablation study with MoireDet+ to examine its best performance module in moiré image identification task on MoireIDT [23] and MRBI [11] dataset with the direct Pixel-sum method (see Table 2). The complete MoireDet+ and ME+SE structures have the best performance. We choose the MoireDet+ to do the following comparison with the four TPDM methods (Pixel-Sum [23], FFT-Based [11], GLCM- and CNN-Based) (see Table 3). The CNN-based method reaches the best performance on both datasets. So we choose MoireDet+ with CNN-Based TPDM method to do the moire image identification with existing other methods: MDCNN [1], Peak [11], Wavelet [1] and MoireDet [23]. Results are in Table 4.

We also need to point out that although MoireDet+ employs the CNN method for detection in Table 4, which performs best overall, other methods may be more suitable in specific cases. For instance, in the MoireIDT dataset, where the CNN method exhibits a lower recall, the GLCM method performs better (81.36 by GLCM vs. 72.10 by CNN). This suggests that different TPDM detection methods can be selected based on the specific requirements of the application, such as a higher emphasis on recall or precision.

Differently, MoireDet+ is more generalized, and achieves the highest precision on the MoireIDT dataset, and has the most stable performance overall. It shows a much larger improvement on the MoireIDT and MRBI datasets than on MoireFace. One possible reason is that the MoireFace dataset was captured on screen conditions that differed from other datasets, causing the moiré pattern to change. A moiré pattern is very well suppressed on miniLED screens, so the thresholds commonly used on LCD screens were used, leading to misclassification.

5.4 Prediction on Continuous Video Frames

When predicting moiré patterns from video, we need to tackle the problem of the model being sensitive to the detail of the input image and give different

Table 4. Performance comparison of moiré pattern identification on three datasets [1, 11,25]. *P*: Precision(%), *R*: Recall(%), *Acc*: Accuracy(%)

	MoireIDT			MoireFace			MRBI		
	P	R	Acc	P	R	Acc	P	R	Acc
MDCNN	50.32	**97.75**	50.63	92.96	88.00	82.77	51.38	**98.53**	52.65
Peak	74.71	79.90	76.43	95.27	**97.33**	**93.08**	73.07	69.41	71.91
Wavelet	38.54	40.20	38.05	76.24	23.00	22.31	53.89	87.65	56.32
MoireDet	84.13	77.90	81.60	**95.88**	93.00	90.57	90.11	89.06	90.00
MoireDet+	**88.5**	72.1	**82.1**	89.13	88.62	88.97	91.08	90.11	**90.42**

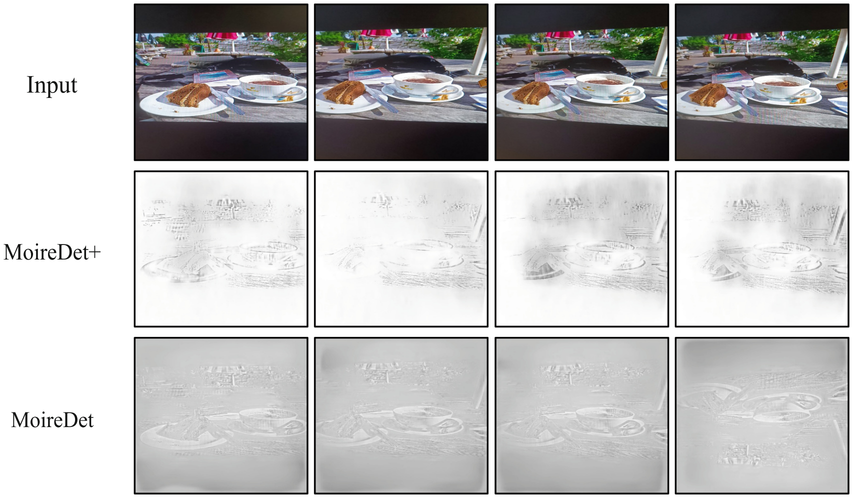


Fig. 6. Visualization on Continuous Video Frames Prediction by MoireDet+ and MoireDet

predictions for continuous video frames that are almost the same. Thus, we compare MoireDet+ on continuous screen-shot video frames with MoireDet (see Fig. 6). We collect a frame-by-frame dataset from several screen-shot videos captured using a 60 FPS (Frame per Second) phone camera, extracting one frame from every five continuous frames. We scale the frames to a size of 320 × 320. Subsequently, both MoireDet and MoireDet+ are applied to these continuous frames. The comparison results are shown in Table 1. Our findings demonstrate a significant improvement in the smoothness of MoireDet+ output compared to MoireDet, with minimal jitter observed on nearly identical image frames. This substantial improvement not only enhances the overall performance of the model but also its stability and robustness.

To quantify the stability of this task, we propose a novel metric called the Continuous Variation Ratio (CVR). For a series of the same frames with iota difference length Len, CVR is defined as:

$$CVR = \frac{\|S_i - E(S)\|_1}{Len \times E(S)} \quad . \tag{12}$$

CVR measures the variability ratio of the output for continuous frames, where S_i represents the sum of t pixel values for frame i, and $E(S)$ denotes the expected value of the sum across all frames. In our experiment, MoireDet+ exhibited a CVR that was 24.5% lower than that of MoireDet, confirming the superior stability of MoireDet+ in continuous frame prediction.

5.5 Effectiveness of Loss Function Evaluation

In Sect. 4.1, we have demonstrated the lower bound of our loss function evaluation algorithm's approximation. Since this method is not a direct quantitative approach, we do not have an effective way to verify its intrinsic validity. Therefore, we present the accuracy results on the MRBI dataset by training MoireDet and MoireDet+ with the loss functions before and after the evaluation, as shown in Table 5. The results show that the model not only achieves improved accuracy after the evaluation but also gains the ability to output based on the specific moiré pattern details in the image, rather than producing a fixed pattern format.

Table 5. Effectiveness of Loss Function Evaluation (Acc(%), test on MRBI)

	MoireDet		MoireDet+	
	MRBI	MoireIDT	MRBI	MoireIDT
Before	90.00	81.60	88.35	80.66
After	90.21	81.93	90.42	82.10

6 Limitations

We encounter the occasional unpredictability of MoireDet+ predictions, particularly notable in the FHDMi dataset, where the output often appears sparse and misleading. We suspect this impunity is caused by the variability in screen technologies and the scaling we employ on the images. In MoireScape, moiré patterns were captured from single-display monitors with different phone cameras. However, our experiments revealed disparities in moiré manifestation across various display types, such as VA (Vertical Alignment), IPS (In-Plane Switching), and TN (Twisted Nematic) monitors. This disparity poses a considerable challenge, as the vast array of monitor parameters can generate diverse moiré patterns.

7 Conclusion

This paper presents MoireDet+, an optimized Moiré Pattern Detection model. By introducing a new type of Vision Transformer inspired by MB-TaylorFormer, enhancing the model structure with a mixed-encoder backbone, and incorporating a contrast detection module, we significantly improve its performance in moiré pattern detection. MoireDet+ demonstrates compatibility with various TPDM methods and exhibits stability and validity in the Moiré Identification Task. Additionally, we introduce an evaluation method for task-specific loss optimization. Our proposed MoireDet+ achieves state-of-the-art performance across several main-stream datasets. For further development and to ensure the reproducibility of our results, we make our codes and evaluations publicly available on https://github.com/Siztas/MoireDetPlus.

Acknowledgements. This work was funded by the National Natural Science Foundation of China (Grant Number: 62473276), the Natural Science Foundation of Jiangsu Province (Grant Number: BK20241918), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant Number: 22KJB520008), the Research Fund of Clobotics Research (Grant Number: KB1801ZW201609-03), and the Research Fund of Horizon Robotics (H230666).

References

1. Abraham, E.: Moiré pattern detection using wavelet decomposition and convolutional neural network. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1275–1279. IEEE (2018)
2. Amidror, I.: Sub-Nyquist Artifacts and Sampling MoirÃ Effects, Part 2: Spectral-Domain Analysis, p. 63 (2015). http://infoscience.epfl.ch/record/206457
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 1–27 (2011)
4. Chen, J., Mai, H., Luo, L., Chen, X., Wu, K.: Effective feature fusion network in BIFPN for small object detection. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 699–703. IEEE (2021)
5. Chen, Z., et al.: Piou loss: towards accurate oriented object detection in complex environments. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, pp. 195–211. Springer (2020)
6. Clough, J.R., Byrne, N., Oksuz, I., Zimmer, V.A., Schnabel, J.A., King, A.P.: A topological loss function for deep-learning based image segmentation using persistent homology. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 8766–8778 (2020)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
8. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
9. Gao, F., Luo, X., Yang, Z., Zhang, Q.: Label smoothing and task-adaptive loss function based on prototype network for few-shot learning. Neural Netw. **156**, 39–48 (2022)

10. Gao, T., Guo, Y., Zheng, X., Wang, Q., Luo, X.: Moiré pattern removal with multi-scale feature enhancing network. In: 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 240–245. IEEE (2019)
11. Garcia, D.C., de Queiroz, R.L.: Face-spoofing 2D-detection based on Moiré-pattern analysis. IEEE Trans. Inf. Forensics Secur. **10**(4), 778–786 (2015)
12. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **6**, 610–621 (1973)
13. 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)
14. Hu, X., Fu, C.W., Zhu, L., Qin, J., Heng, P.A.: Direction-aware spatial context features for shadow detection and removal. IEEE Trans. Pattern Anal. Mach. Intell. **42**(11), 2795–2808 (2019)
15. Kingma, D.P., Ba, J.: Adam: a Method for Stochastic Optimization (2014). arXiv preprint arXiv:1412.6980
16. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer (2014)
17. Luo, Q., Li, H., Chen, Z., Li, J.: Add-UNET: An adjacent dual-decoder UNET for SAR-to-optical translation. Remote Sens. **15**(12), 3125 (2023)
18. Makarau, A., Richter, R., Müller, R., Reinartz, P.: Haze detection and removal in remotely sensed multispectral imagery. IEEE Trans. Geosci. Remote Sens. **52**(9), 5895–5905 (2014)
19. Mohanaiah, P., Sathyanarayana, P., GuruKumar, L.: Image texture feature extraction using GLCM approach. Int. J. Sci. Res. Publ. **3**(5), 1–5 (2013)
20. Mustafa, A., Mikhailiuk, A., Iliescu, D., Babbar, V., Mantiuk, R.: Training a task-specific image reconstruction loss. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 21–30 (2022)
21. Niu, Y., Lin, Z., Liu, W., Guo, W.: Progressive Moire removal and texture complementation for image demoireing. In: IEEE Transactions on Circuits and Systems for Video Technology (2023)
22. Qiu, Y., Zhang, K., Wang, C., Luo, W., Li, H., Jin, Z.: Mb-taylorformer: multi-branch efficient transformer expanded by Taylor formula for image dehazing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12802–12813 (2023)
23. Yang, C., Yang, Z., Ke, Y., Chen, T., Grzegorzek, M., See, J.: Doing more with Moiré pattern detection in digital photos. IEEE Trans. Image Process. **32**, 694–708 (2023)
24. Yue, H., Cheng, Y., Liu, F., Yang, J.: Unsupervised Moiré pattern removal for recaptured screen images. Neurocomputing **456**, 352–363 (2021)
25. Yue, H., Mao, Y., Liang, L., Xu, H., Hou, C., Yang, J.: Recaptured screen image demoiréing. IEEE Trans. Circuits Syst. Video Technol. **31**(1), 49–60 (2020)

Rare Fungi Image Classification Based on Few-Shot Learning and Data Augmentation

Jiayi Hao[1], Yulin Feng[1], Wenbin Li[2], and Jiebo Luo[1](✉)

[1] University of Rochester, Rochester, NY 14627, USA
jluo@cs.rochester.edu
[2] Nanjing University, Nanjing, People's Republic of China

Abstract. Fungi image classification is highly challenging due to the high degree of similarity in the visual features and varying image quality. The classification of rare fungi is made more difficult by the limited data and images available. To address this issue, we employ few-shot learning techniques on the Danish Fungi 2020 dataset, utilizing the LibFewShot implementation. In particular, our study focuses on experimenting with five metric-learning based few-shot learning methods and comparing their performances on this dataset. Further, we examine the effectiveness of applying five data augmentations on each method, and find that adding all beneficial augmentations does not yield better results than applying the most beneficial augmentation alone. We also attempt to enhance the models with two self-supervised learning tasks, where we discover them to have the best performance on weaker models. Similarly, when adding augmentations to self-supervised tasks, the overall performance was weakened. Overall, we have found the Cross Attention Network with ColorJitter augmentation to be the optimal model in this application along with a remarkable scalability. Our study provides insights into the potential of utilizing few-shot learning to classify uncommon fungi and directions for further improvements.

Keywords: Fungi Image Classification · Few-Shot Learning · Data Augmentation · Self-Supervised Learning Tasks

1 Introduction

The classification of fungi has been identified as a crucial component in biotaxonomy, offering researchers a deeper understanding of biodiversity within mycology [16]. Employing fungi image classification models presents a considerable time-saving alternative to traditional manual classification, which also requires less human expertise. Furthermore, fungi recognition is the core technique in fungi recognition apps that are emerging for people interested in fungi, similar to other smartphone apps for flowers and plants. Despite the critical role such models could play, the explorations in fungi image classification have been

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 50–62, 2025.
https://doi.org/10.1007/978-3-031-78444-6_4

relatively limited, primarily due to the absence of relevant datasets. This scenario is improved by the introduction of the Danish Fungi 2020 Dataset (DF20) [12], marked by its unparalleled quality and scale. The dataset, which originated from the Atlas of Danish Fungi, encompasses 1,604 classes with 266,344 training instances and establishes an ideal test bed for conducting image classification experiments related to fungi.

By employing data in DF20 and state-of-the-art (SOTA) image classification methods, the majority of fungi image classification tasks—except for rare species—should be significantly more manageable. Given the scarcity of available data for these particular species, ordinary deep-learning methodologies prove ineffective for their classification. However, few-shot Learning (FSL) [3,18] can be employed to address this challenge. FSL enables classifiers to be first trained on an abundant but mutually exclusive auxiliary dataset (e.g., common fungi species), enabling the transfer of this knowledge to the classification of rare species. Currently, three primary categories of FSL methods exist: meta-learning based methods [5], metric-learning based methods [14,17], and non-episodic based methods [2]. LibFewShot [8] provides a comprehensive library of eighteen representative state-of-the-art FSL methods. Utilizing LibFewShot's implementation, FSL experiments can be conducted with ease and efficiency. Despite FSL methods being widely experimented on other popular benchmark datasets like miniImageNet and tieredImageNet, to our knowledge, only one work has applied FSL to the DF20 dataset. [1] criticized current FSL benchmarks as "far from real case" and proposed DF20 as a better benchmark. However, their work mainly focused on improving the current benchmark with a better sampling technique while the experiments on DF20 were limited.

In this study, we conduct a comprehensive experimental analysis on the DF20 dataset with current SOTA metric-learning based FSL methods. In our experiments, we first propose a data split on the DF20 dataset according to the conventional FSL setting to conduct a rare fungi classification task. Then, we test five state-of-the-art metric-learning based FSL methods on the 5-way 5-shot tasks and compare the results with other benchmarks. We also explore the effects of five data augmentations and two self-supervised learning (SSL) tricks [6,15] as regularization in these methods. Further, experiments are conducted on the 20-way 5-shot tasks with selected FSL methods. Finally, we combine the best-performed methods, regularization, and SSL tricks in this classification task. By adding self-supervised tasks to the metric-learning based FSL methods in the LibFewShot environment, we also enable a fair comparison of the effectiveness of self-supervised tricks in metric-learning based FSL methods. We hope our study can provide insights into employing the few-shot learning method to address the rare fungi image classification challenge and other similar domains.

The main contributions of this work are as follows:

- We investigate few-shot learning techniques for the challenging problem of rare fungi image classification.

- We examine the effectiveness of employing various data augmentation and self-supervised learning strategies in conjunction with few-shot learning to improve classification performance.
- We provide insights into using few-shot learning to classify uncommon fungi and achieve respectable accuracy that can facilitate practical.

2 Related Work

Picek *et al* proposed the first large-scale fungi dataset, i.e., Danish Fungi 2020. They proposed this data as a new benchmark and discussed the uniqueness of this dataset which we will describe later in Sect. 3. The researchers also provided baseline performance results using Convolutional Neural Network (CNN) and Vision Transformer, which showed the DF20 dataset is challenging.

Li *et al.* proposed and built a comprehensive library for few-shot learning called LibFewShot by re-implementing eighteen SOTA few-shot learning methods in a unified framework. In the paper, they first reviewed the methods of few-shot learning, including non-episodic based, meta-learning based, and metric-learning based, and then compared the performance of 18 state-of-the-art models on miniImageNet and tieredImageNet datasets.

Bennequin *et al.* published a critique on few-shot learning methods arguing that the old benchmarks (i.e., miniImageNet and tieredImageNet) are too far from the "real industrial cases", and suggested DF20 as a more challenging and better benchmark for FSL. In their paper, they compared the performances of six FSL methods on tieredImageNet using DF20 as a testing set and concluded that the FSL models performed much worse in DF20. Therefore, they suggested researchers use more many-way classification, like in DF20, when evaluating FSL methods.

Su *et al.* scrutinized the impact of self-supervised learning (SSL) on few-shot learning performance. They explored the utility of SSL on small datasets, a subject relatively untouched in prior studies. They emphasized that traditional training methods might discard important semantic information when the focus is solely on base class classification. To address this, they advocated for leveraging SSL as an auxiliary task to improve the model's ability to generalize to new classes in a few-shot scenario. Their findings suggested that SSL can reduce the relative error rates of few-shot learners, particularly when dealing with small or challenging datasets.

Recent advances in the FSL field have also highlighted several important contributions. Notably, [13] proposed the Contrastive Language-Image Pre-Training (CLIP) model, a vision-language model pre-trained on a massive dataset of 400 million image and text pairs. The CLIP model demonstrated remarkable zero-shot capabilities, accurately predicting the descriptive text for provided images with performance comparable to ResNet50, without task-specific optimization. Building on the success of CLIP, [20] introduced Large Language Models as Prompt Learners (LLaMP) and leveraged the capabilities of large language models to enhance CLIP. The LLaMP model showed significant improvements across

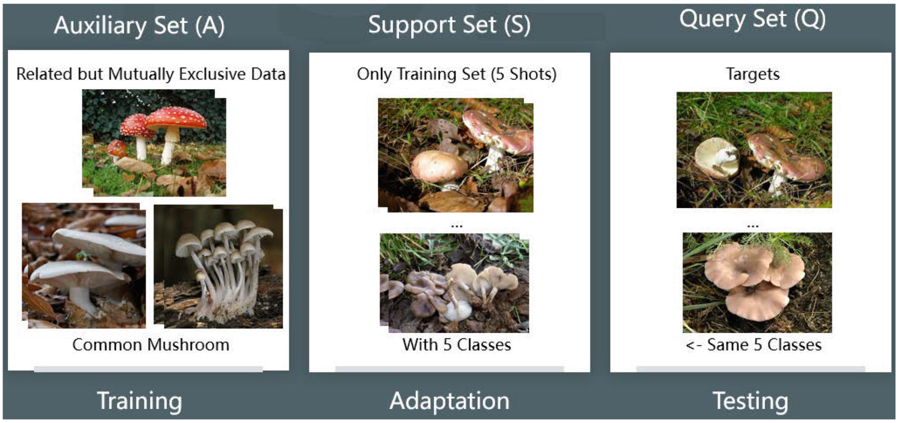


Fig. 1. Example fungi images in DF20

11 datasets in both zero-shot and few-shot scenarios. Additionally, research such as [11] has provided critical insights into the robustness of FSL, underscoring the importance of addressing security issues of FSL models, especially in their real-world applications.

While the fungi data and the FSL methodology exist in previous works, no study has thoroughly examined the effects of combining them. Therefore, we conduct a comprehensive experimental study aiming to explore the possibility of using FSL methods to solve the challenging problem of fungi image classification. By attempting to build an effective classification model, we hope to inspire the future development of practical solutions.

3 Dataset

3.1 Data Preparation

Following the common settings of the few-shot learning datasets [18], we split the dataset into three parts: training, validation, and testing. In fact, in each training/test epoch, many training/testing episodes will be built from these sets. In the training stage, the training set will be used as the Auxiliary set, and training episodes will be built from it. Similarly, in the testing stage, the Support set and Query set will be constructed from the testing set. The validation set is used, after each training epoch, to validate that the parameter updates can be generalized to other classes by calculating the accuracy on a separate set. This process is employed to avoid over-fitting and ensure the model can adapt to the target Query classes. Notably, it is important to ensure that the classes in training, validation, and testing sets must be mutually exclusive. Illustrative examples of dataset images are depicted in Fig. 1.

For the Danish Fungi 2020 (DF20) dataset with 1,604 distinct classes, a thresholding approach is used to separate rare species: classes with fewer than

40 observations are identified as rare and thus split into the testing set. This threshold ensures that approximately 10% of the total classes are used for testing purposes. This threshold should be chosen considering the trade-off of model accuracy and generalizability. Specifically, a higher threshold will include more fungi species in the testing set, allowing the model to be more practical but also making the training process much more challenging.

Table 1. Data Split on the DF20 Dataset

Subset	Number of Classes	Number of Images
Training	1270	260506
Validation	148	29037
Testing	186	6398
Total	1604	295941

The remaining 90% of classes go through a random sampling process, with a stipulated 10% of these classes allocated to the validation set. As summarized in Table 1, this data split results in an unbalanced distribution of classes, noted by a higher number of classes in the testing set. This distribution, though unconventional to standard practices that include a higher number of classes in the training set, is particularly tailored to suit the objective of rare fungi classification. This data split emphasizes the challenge of classifying lesser-documented species, thereby facilitating a more rigorous examination of model performance within the specific context of this study.

4 Methodology

4.1 Problem Formulation

In a few-shot learning problem, there are three sets of data: the Auxiliary set (A), the Support set (S), and the Query set (Q). The “few-shot” concept refers to the number of samples available in the Support set S.

In our application, the rare fungi classes we aim to classify are in the Support set and Query set, while other common classes are in the Auxiliary set. The Auxiliary set enriches the model with extra classes and samples. To better utilize extra data, Few-shot learning models aim to propose better adaptation methods to overcome the domain difference between the Auxiliary and Query sets. These different adaptation methods primarily differentiate various few-shot learning algorithms.

4.2 Metric-Learning Based Methods

In this study, we opt for metric-learning based FSL methods because [8] revealed that the test-tuning during the test stage is not necessary for few-shot learning due to the limited amount of data. Consequently, choosing metric-learning based methods, which avoid the need for test-tuning, appears to be reasonable. A metric-learning based model has two major components - an embedding backbone and a classifier. These two components serve different purposes: the backbone is a Convolutional Neural Network that learns how to extract features from the images, while the classifier learns how to classify based on the features.

Metric-learning based methods focus on comparing classes through learning better measurements of similarity and dissimilarity between images and classes, i.e. metric functions. There are various methods in this division while the methods we experiment on are Prototypical Networks (ProtoNet) [14], Relation Network (RelationNet) [17], Covariance Metric Networks (CovaMNet) [9], Deep Nearest Neighbor Neural Network (DN4) [7], and Cross Attention Network (CAN) [19]. We briefly describe the characteristics of these methods as follows.

ProtoNet [14], one of the earliest and most classic metric-learning based methods, calculates the Euclidean distances between the Query image feature vector and the mean feature vector of each Support class (i.e. class prototypes), and makes classifications with a 1-Nearest-Neighbor classifier. For RelationNet [17], instead of using a fixed metric, a relation network is employed as the classifier which learns a metric dynamically using the training stage. The learned metric is then used to calculate a similarity score for classification. CovaMNet [9], instead of using first-order distance calculation, employs a covariance metric layer to calculate the second-order similarity metric and then make classification with a softmax function. DN4 [7], improving on ProtoNet, directly uses a *local descriptor* to calculate the similarity between images and classes, instead of first pooling down the features with fully connected layers. Finally, CAN [19], approaches the problem by understanding what to compare between target Query images and Support images. After embedding the backbone, CAN includes a Cross Attention Module which highlights the objects-to-compare in both Query and Support images by calculating the class attention map, and thus produces the final features that have higher discriminative power.

4.3 Data Augmentation and Self-supervised Learning Tasks

Based on the FSL methods, we are interested in further examining if the model performance can be improved by applying proper data augmentation and/or self-supervised learning tasks. The key motivation is to enhance the model's ability to extract semantic information (characteristics of fungi). This is particularly important because the majority of the fungi images are taken outdoors: under different lights, of different distances and positions. However, the available training sample is insufficient to cover these variations. Therefore, we hope proper data augmentation can help the model generalize better to these unseen conditions and thus avoid over-fitting.

The use of self-supervised learning tasks shares a similar purpose. By training the model to solve Jigsaw and Rotation puzzles, we hope to enhance its ability to recognize fungi placed in tilted positions or off the center of the image. In addition to the original FSL framework where labeled images are embedded and used to

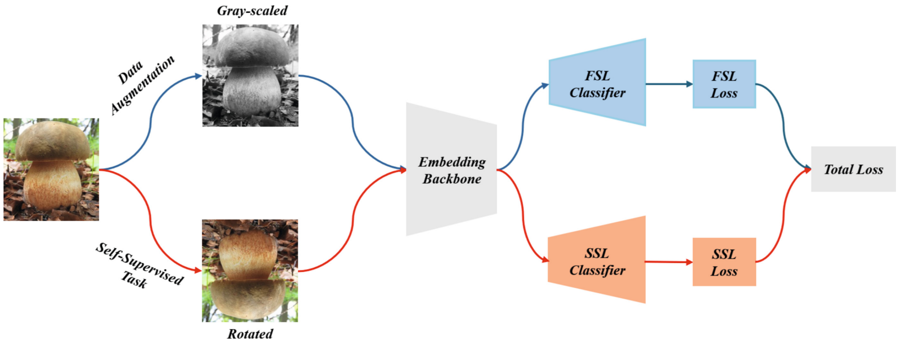


Fig. 2. Pipeline figure of the method

calculate a supervised loss L_s, an SSL data loader and an SSL classifier are added to the model following the framework proposed in [15]. During training, unlabeled images are generated by the SSL data loader according to the specific SSL task, embedded by the same backbone, and passed to the SSL classifier to calculate a self-supervised loss L_{ss}. The final loss function combines supervised and self-supervised losses as $L = (1 - \lambda)L_s + \lambda L_{ss}$, where λ here is a hyperparameter that controls the weights of the two losses. The two self-supervised tasks we experiment with are the Jigsaw task and the Rotation task [6,15]. Each task has a different data loader and classifier. To improve the clarity, we summarize our method as a pipeline in Fig. 2.

5 Experiments

5.1 Experiment Settings

In our experiment, we mainly follow the experiment settings used in LibFewShot and other previous works [8,15].

- **Dataset.** Following the procedure, we split the DF20 dataset into training, validation, and testing sets. The classes are mutually disjointed in these sets. All images are resized to 96 × 96 and RandomCrop to 84 × 84.
- **Embedding Backbone.** We experiment with two embedding backbones—Conv64F [14,18] and ResNet18. Conv64F is adopted by the original ProtoNet implementation which has 4 convolution blocks and 64 filters in each block [14]. ResNet18 is a much deeper network with 17 convolutional layers. The parameter sizes of methods with these backbones can be found in the result in Table 2. Conv64F produces much lower accuracy than ResNet18 and thus its results are omitted.
- **Classifiers.** When having ResNet18 as the embedding backbone, DN4, RelationNet, CovaMNet, and CAN use features without adaptive average pooling or flattening. RelationNet and CovaMNet use stride = 1 in the last convolutional layer. DN4 uses Topk = 3.

- **Training Stage.** All methods are trained for 100 epochs in the training stage. Each epoch includes 600 training episodes (i.e., a total of 6000 tasks). Model updates are saved when accuracy in the validation set is improved. All models are trained with either the ADAM optimizer with an initial learning rate of 0.001 or SGD with an initial rate of 0.1. For details see Table 2.
- **Testing Stage.** All methods are tested for 6 epochs with 600 testing episodes (i.e., a total of 3000 tasks). All accuracy reported is top-1 accuracy (accuracy@1).
- **Data Augmentation.** In the training stage, images are only resized and randomly cropped (i.e. no augmentation). Augmentations we experimented with are Brightness, Contrast, and Saturation in ColorJitter, RandomGrayscale, and HorizontalFlip. In the validation and test stages, all images are center-cropped only.

Table 2. Overview of FSL models' performances. All models use ResNet18 as the embedding backbone

Method	Learning Rate/Optimizer/lr Scheduler	Param. Size	Training Time	Accuracy@1
ProtoNet [14]	0.001/Adam/StepLR	11.17M	3:16:58	83.71
RelationNet [17]	0.001/Adam/StepLR	18.29M	3:24:51	76.56
CovaMNet [9]	0.001/Adam/StepLR	11.17M	3:12:26	70.16
DN [7]	0.001/Adam/StepLR	11.17M	3:18:04	83.98
CAN [19]	0.1/SGD/Cosine	11.82M	2:59:40	87.87

- **Self-supervised Tasks.** For training with self-supervised tasks, namely Jigsaw and Rotation, images are prepared following the procedure in [6,15]. In the SSL classifier, the embedded features from the backbone are passed to several fully connected layers, following the architecture in [15]. For methods that do not use average pooling and flattening in the backbone, we add these layers manually in the SSL classifier.

5.2 Overview of FSL Methods on DF20

In this section, we primarily experiment with the baseline models with ProtoNet [14], RelationNet [17], CovaMNet [9], DN4 [7], and CAN [19] as the FSL method and ResNet18 as the embedding backbone. All the experiments are done in the 5-way 5-shot setting, and no data augmentations are used except for resizing. We evaluate the methods by comparing their top-1 accuracies, model parameter sizes, and training times. The comparison of the models is summarized in Table 2.

The results indicate that CAN outperforms all other methods with an accuracy of 87.87% and the shortest training time. Notably, we discover that CAN requires using SGD as the Optimizer. Adam works well with other methods but results in an accuracy of merely 30% on CAN. The reason for CAN's outstanding performance might be its effective employment of the attention mechanism.

Table 3. Effects of various data augmentation on each method. Accuracy with improvement compared to no augmentation is shown in **bold**. All Positive includes all augmentations with positive effects. The best combination is shown with underline

Method	Without Augment	Brightness	Contrast	Saturation	Gray scale	Horizontal Flip	All Positive
ProtoNet [14]	83.71	**83.89**	**83.79**	**83.79**	80.90	**83.87**	83.88
RelationNet [17]	76.56	20.00	20.00	75.68	73.94	**77.07**	77.04
CovaMNet [9]	70.16	65.82	**72.37**	71.02	**70.98**	**72.92**	72.85
DN4 [7]	83.98	81.79	83.47	**83.99**	80.30	**84.29**	84.11
CAN [19]	87.87	**88.05**	82.82	**88.40**	82.85	82.90	83.71

The Cross Attention Module allowed CAN to learn which parts of the fungi (e.g. cap skirt, stem, or volva) are more important to focus on when classifying fungi species. This could be particularly beneficial in difficult cases where the two species are in the same fungi family and share most appearances. Additionally, distinguishing these cases requires high human expertise, which further amplifies the practical significance of having a fungi image classification model.

When comparing to the benchmarks in the LibFewShot paper [8], we have not found the dataset to be more challenging based on our data-splitting. However, it is possible to make the task more difficult by raising the number of classes in the testing set as there is no currently established train-test-split for DF20.

5.3 Data Augmentation

The augmentation techniques affect the accuracy of few-shot learning models differently, as demonstrated in Table 3. We observe that different methods lead to various responses to the augmentations. For example, ProtoNet consistently demonstrates slight improvement with most augmentations, except for Random Grayscale. Conversely, RelationNet showes extreme sensitivity to brightness and contrast adjustments, resulting in a significant decrease in accuracy. Regarding specific augmentations, HorizontalFlip appears to be highly beneficial across most methods, particularly in enhancing the performances of RelationNet, DN4, and CovaMNet.

The different effectiveness might be influenced by the diverse angles and lighting conditions under which the photos were taken. While these findings could provide valuable insights into selecting beneficial augmentation methods for this dataset and similar scenarios, such as fungi recognition apps, they should not be generalized to all fungi classification tasks, since the augmentations are customized to suit specific photography conditions.

Notably, when combining all augmentations with positive effects into the “All Positive” column, the overall gain is lower than the single most effective augmentation across all methods. This finding suggests that the combination of multiple augmentations is not linearly addictive with performance improvement in this task. It implies that certain augmentations, when applied simultaneously, may diminish each other’s effectiveness.

Table 4. Further results on 20-way 5-shot classification with ProtoNet and CAN methods. Accuracy with improvement compared to no augmentation is shown in **bold**

Method	Without Augment	Brightness	Contrast	Saturation	Grayscale	Horizontal Flip
ProtoNet [14]	71.27	**71.72**	71.11	70.39	66.12	**74.02**
CAN [19]	78.56	78.46	**78.67**	78.36	66.89	**80.09**

Further, we select two well-performed methods—ProtoNet and CAN— and conduct some initial experiments on their performance on 20-way classification. Although 20-way classification is typically a much more challenging task than 5-way, we notice from Table 4 that the transition from 5-way to 20-way classification does not significantly diminish accuracy, especially in CAN. This result suggests a surprising scalability of these methods in the Fungi classification task. In addition, the comparison of the 5-way and 20-way results indicates that the effectiveness of the augmentations varies when the number of classes changes. For example, HorizontalFlip is more effective in the 20-way setup, while other augmentations that benefit the 5-way scenario do not translate into the same level of improvement in the 20-way setting. This observation emphasizes the importance of optimizing the augmentation methods separately for tasks of different way numbers.

5.4 Self-Supervised Learning Tasks

Does solving self-supervised learning puzzles help? In our experiments, we experiment on two SSL tasks—Jigsaw and Rotation. To keep a fair comparison, we maintain the same model framework in LibFewShot, and add supports to the self-supervised learning tasks by introducing additional data-loading procedures and classifiers in the original metric-learning methods, following the architecture in [15].

Table 5. Effects of self-supervised tasks on each method. Each self-supervised task is tested with loss weight λ equals 0.2 or 0.5. Accuracy with improvement compared to no SSL task is shown in **bold**. Augmentation+SSL shows the result using the best data augmentation plus SSL task, while "-" denotes when no SSL task is beneficial

Method	Without	Jigsaw		Rotation		Augmentation+SSL
		$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.2$	$\lambda = 0.5$	
ProtoNet [14]	83.71	80.72	81.14	**83.81**	**84.03**	84.00
RelationNet [17]	76.56	74.76	74.01	**76.97**	**78.57**	76.57
CovaMNet [9]	70.16	68.62	**73.98**	65.60	**73.27**	70.36
DN4 [7]	83.98	83.10	82.32	81.89	83.24	–
CAN [19]	87.87	78.18	80.10	70.77	73.89	–

In contrast to the improvements presented in [15] with ProtoNet, SSL tasks yield benefits primarily for weaker methods in our experiment, as shown in Table 5, with the highest improvement of over 3%. Specifically, we find the Rotation task beneficial for ProtoNet, although the improvements are less pronounced compared to those reported in [15].

However, no improvement is observed for stronger models like CAN. This could be attributed to the loss of spatial information, which is essential for the attention mechanism in CAN, during the Jigsaw and Rotation operations. This finding is consistent with the results in Table 3, where the HorizontalFlip augmentation is not beneficial for CAN in the 5-way setting. This further illustrates that augmentations related to spatial information should be used with caution for CAN.

More importantly, we find that the selection of the λ parameter significantly influences the efficacy of SSL methods in our experiments. As illustrated in the Table 5, for CovaMNet, varying λ choices lead to effects ranging from an improvement of over 3% to a reduction of over 4%. Similar patterns are observed across all methods, emphasizing that λ needs to be carefully tested and selected for each method to optimize results.

We also experiment with the effects of combining data augmentations and SSL tasks for those methods in which SSL tasks are beneficial, i.e., ProteNet, RelationNet, and CovaMNet. Interestingly, we observe a reduction in accuracy across all three methods when incorporating the previously identified beneficial data augments outlined in Sect. 5.3 into the top-performing SSL tasks. This finding, coupled with the findings from data augmentation, initially suggests that adding extra augmentations might not be as beneficial as simply choosing the most effective one.

Combining the findings from Sects. 5.3 and 5.4, we can summarize the most beneficial augmentation and SSL task combinations for each metric-learning based method in the 5-way 5-shot classification task. The results, along with their corresponding accuracies, are presented in Table 6. Overall, the optimal solution for this particular task emerged as CAN with saturation alone, achieving an accuracy of 88.4%. For the 20-way 5-shot task, the best solution is CAN with HorizontalFlip, achieving a remarkably satisfactory accuracy of 80.09%.

Table 6. Best combination of methods, data augmentation, and self-supervised task

Method	Combination	5-way 5-shot Acc@1
ProtoNet [14]	Rotation ($\lambda = 0.5$)	84.03
RelationNet[17]	Rotation ($\lambda = 0.5$)	78.57
CovaMNet [9]	Jigsaw ($\lambda = 0.5$)	73.98
DN4 [7]	HorizontalFlip	84.29
CAN [19]	Saturation	88.40

6 Conclusion

By experimenting with state-of-the-art metric-learning based few-shot learning methods on the Danish Fungi 2020 dataset, we have found the Cross Attention Network (CAN) as the best-performing method, demonstrating the highest accuracy and reasonable training time. CAN also exhibits remarkable scalability in 20-way classification tasks. For data augmentation, HorizontalFlip is beneficial across metric-learning methods except for CAN. When integrating self-supervised learning techniques, positive effects are primarily observed in weaker models, such as ProtoNet. The loss weight λ significantly influences the outcomes and thus requires careful selection.

While this study offers valuable insights, there is ample room for further research. CAN has outperformed other methods in this specific task, but augmentations and self-supervised learning (SSL) tasks fail to enhance its accuracy significantly. Future studies should explore additional augmentation methods or self-supervised learning tasks that preserve spatial information to refine CAN further. Exploring other SSL beyond the Jigsaw and Rotation tricks is also useful. Regarding SSL, although we have not been able to thoroughly experiment with the choice of λ, it is worth tuning it for each FSL model separately. Additionally, investigating the potentials of the latest FSL methods, such as [10] and [4], could yield valuable findings. Finally, it would be interesting to develop a smartphone app for fungi recognition and compare it with the limited offerings currently on the smartphone app market.

Acknowledgment. This research is supported in part by the Goergen Institute for Data Science at the University of Rochester.

References

1. Bennequin, E., Tami, M., Toubhans, A., Hudelot, C.: Few-shot image classification benchmarks are too far from reality: Build back better with semantic task sampling. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4766–4775 (2022)
2. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C., Huang, J.B.: A closer look at few-shot classification. In: International Conference on Learning Representations (2019)
3. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 594–611 (2006). https://doi.org/10.1109/TPAMI.2006.79
4. Fifty, C., Duan, D., Junkins, R.G., Amid, E., Leskovec, J., Re, C., Thrun, S.: Context-Aware Meta-learning (2024). https://arxiv.org/abs/2310.10971
5. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning (ICML), pp. 1126–1135 (2017)
6. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P.P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8058–8067 (2019). https://doi.org/10.1109/ICCV.2019.00815

7. Li, W., Wang, L., Xu, J., Huo, J., Gao, Y., Luo, J.: Revisiting local descriptor based image-to-class measure for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7260–7268 (2019)
8. Li, W., Wang, Z., Yang, X., Dong, C., Tian, P., Qin, T., Jing, H., Shi, Y., Wang, L., Gao, Y., Luo, J.: LibFewshot: a comprehensive library for few-shot learning. IEEE Trans. Pattern Anal. Mach. Intell. **01**, 1–18 (2023)
9. Li, W., Xu, J., Huo, J., Wang, L., Gao, Y., Luo, J.: Distribution consistency based covariance metric networks for few-shot learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8642–8649 (2019)
10. Lim, J.Y., Lim, K.M., Lee, C.P., Tan, Y.X.: SSL-protonet: self-supervised learning prototypical networks for few-shot learning. Expert Syst. Appl. **238**, 122173 (2024). https://doi.org/10.1016/j.eswa.2023.122173
11. Liu, X., Jia, X., Gu, J., Xun, Y., Liang, S., Cao, X.: Does few-shot learning suffer from backdoor attacks? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 19893–19901 (2024)
12. Picek, L., et al.: Danish fungi 2020 - not just another image recognition dataset. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1525–1535 (January 2022)
13. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
14. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
15. Su, J.C., Maji, S., Hariharan, B.: When does self-supervision improve few-shot learning? In: European Conference on Computer Vision (ECCV), pp. 645–666. Springer (2020)
16. Sulc, M., Picek, L., Matas, J., Jeppesen, T., Heilmann-Clausen, J.: Fungi recognition: a practical use case. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2020)
17. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199–1208 (2018)
18. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
19. Xiao, B., Liu, C.L., Hsaio, W.H.: Semantic cross attention for few-shot learning. In: Proceedings of the 14th Asian Conference on Machine Learning (ACML), vol. 189, pp. 1165–1180 (2023)
20. Zheng, Z., Wei, J., Hu, X., Zhu, H., Nevatia, R.: Large language models are good prompt learners for low-shot image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 28453–28462 (2024)

Interpretative Attention Networks for Structural Component Recognition

Abhishek Uniyal[1], Bappaditya Mandal[2](✉), Niladri B. Puhan[1], and Padmalochan Bera[1]

[1] School of Electrical Sciences, Indian Institute of Technology, Bhubaneswar, India
{21cs06019,nbpuhan,plb}@iitbbs.ac.in

[2] School of Computer Science and Mathematics, Keele University, Keele, UK
b.mandal@Keele.ac.uk

Abstract. Bridges are essential for enabling movement during environmental disasters and serve as crucial links for rescue and aid delivery. Effective bridge inspection and maintenance are more critical than ever due to increasing severity and frequency of environmental disasters. Although current state-of-the-art deep learning models have achieved good performance many challenges still exist, such as their performance on challenging datasets and their opaque-box nature makes it difficult to understand their decision-making process and identify potential biases. This research work proposes a novel architecture that incorporates innovative parallel twin attention module, synchronous amplification module, aggregated multi-feature attention module and squeeze and excitation blocks, that helps to focus on specific regions of the image plane automatically resulting in improved structural component recognition accuracy. Its parallelism helps to capture long-range dependencies enabling the model to use contextual information encompassing spatial and channel information when segmenting bridge components. Experimental results and ablation studies show that our proposed architecture outperforms the current state-of-the-art methodologies in the challenging bridge component classification dataset. We also examine our models through XAI methods to provide insights into its decision-making process and making it more trustable by highlighting the importance of different features for various similar recognition/segmentation tasks.

Keywords: Attention Networks · Structural Component Recognition · Interpretative Models

1 Introduction and Background Work

In emergency situations, critical structures like bridges and other civil structures play a crucial role in ensuring public safety, facilitating emergency response efforts and enhancing overall resilience. These structures can help in evacuation routes, access for emergency services, supply chain support and post-disaster

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 63–77, 2025.
https://doi.org/10.1007/978-3-031-78444-6_5

recovery. Due to this, it has become highly important to maintain these infrastructure and also monitoring their health. By investing in regular inspections, repairs, and upkeep, communities can enhance safety, reduce long-term costs, maintain operational continuity, and improve the overall resilience of their critical infrastructure. Achieving these tasks by manual inspection poses significant challenges such as limited accessibility, time consuming/criticality, safety risk for humans, disruptions to normal traffic, real time monitoring and many others.

Computer vision (CV) and machine learning (ML) can be helpful in bridge and other structure inspections by automating image analysis, improving accuracy and consistency, enabling remote and difficult access inspections, providing real-time monitoring, facilitating data-driven decision making, and analysing historical data for insights into structural behaviour and deterioration [2–4,7,22]. A key initial step in the automated inspection of civil infrastructure is expected to be the method of detecting and recognising distinctive portions of a structure using structural component recognition (SCR). SCR is the crucial initial step towards assessing the health of structures as it eliminates interference caused by damage or other non-structural components and leads to more accurate localisation and/or improves the usefulness/practicality of the system [7,18]. Recognising these components using image/video data is very challenging task due to degradation of components, environmental conditions such as shadows or lightning which affect the image data, variability and occlusion by other objects and a multitude of other problems [2,10,13,14,21].

Narazaki et al. [14] proposed a method that combines multiscale convolutional neural networks (CNNs) to achieve SCR, where the approach is inspired by [5] for hierarchical scene labelling in images. Their integration with scene classification helped to minimise false-positive results and ensured consistent labels. Furthermore, Narazaki et al. [13] extended the approach by employing the fully convolutional network (FCN) and SegNet architectures. Two variations of the SegNet architecture, namely SegNet45-S and SegNet45, are defined based on the number and sizes of layers after upsampling. Three network configurations, namely naive, parallel, and sequential, are employed for each network architecture. These configurations are used to combine the scene understanding and bridge component classification, enhancing the overall recognition process. Recently attention networks has emerged as a powerful tool within the framework of CNN for SCR [10,21], offering significant advantages over traditional CNNs alone. The attention mechanism enables the model to concentrate on particular regions of the input that are regarded more essential or informative as opposed to CNNs, which process the full input uniformly.

2 Interpretative Attention Network

The proposed interpretative attention network (IAN) incorporates attention mechanisms at various levels, which captures intricate details, correlations, spatial dependencies for accurate recognition of diverse components within complex structures and their ambiguous continuations [7,18]. With parallel twin attention module (PTAM) at its core, along with other attention mechanisms it facilitates the extraction of high-level representations that capture the distinct characteristics of different structural components as shown in Fig. 1.

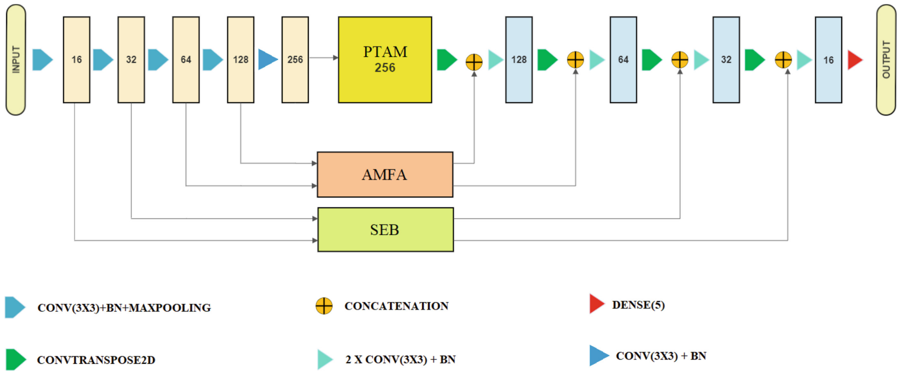


Fig. 1. Architecture of the proposed IAN model for structural component recognition. It consists of mainly 3 attention modules PTAM, aggregated multi-feature attention module (AMFA) and squeeze and excitation block (SEB). In the first stage, inputs passes through convolution layer of 3×3 kernel size, with number of filters represented inside each intermediate block, followed by batch normalisation and max pooling. In second stage, outputs from attention layers is passed through convolution transpose layers followed by concatenation with corresponding layers from first stage using skip connection and then the convolution layers of 3×3 kernel size with batch normalisation. Lastly, dense layer is added which have 5 nodes.

2.1 Parallel Twin Attention Module (PTAM)

The PTAM consists of multiple attention operating simultaneously/in parallel as shown in Fig. 2(a), this involves spatial and channel attentions and aggregation of their results to extract discriminative features for multiple target structural component classification task. The parallel twin operation of PTAM is achieved by using convolutional block attention and synchronous amplification modules.

Convolutional Block Attention Module (CBAM): We have adapted CBAM from [24] by incorporating innovative channel attention module (CAM) and the spatial attention module (SAM), to improve the representational strength of traditional convolutional neural networks as shown in Fig. 2(b). On the feature maps, the CAM logs how different channels are interdependent on one another. To obtain a channel descriptor, pooling operation is performed, which includes both max and average pooling, whose description features include the channel's general statistics. Following that, two fully connected layers with rectified linear unit (ReLU) activation are used to simulate the channel-wise relationships. These layers learned how to create weights for channel attention as is evident from the experimental and ablation studies.

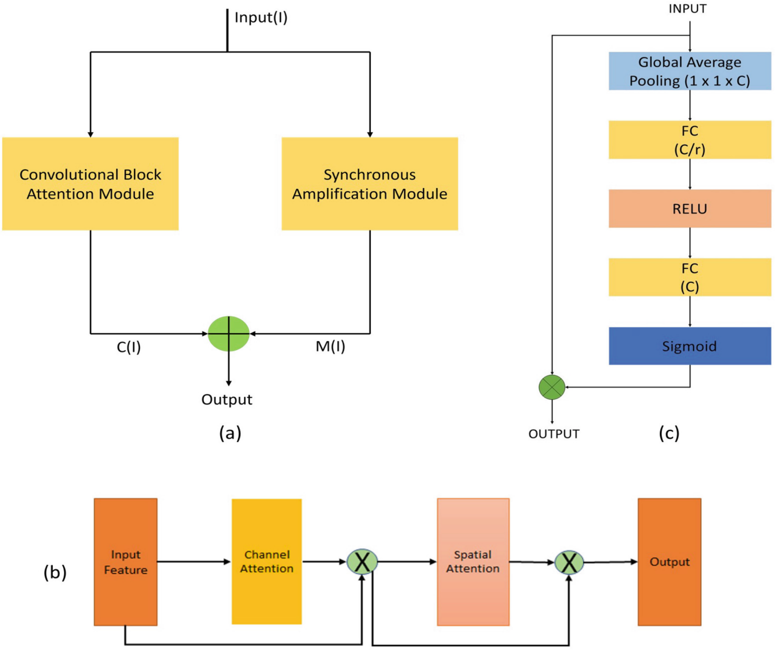


Fig. 2. (a) Parallel twin attention module (PTAM) incorporating convolutional block attention module (CBAM) and synchronous amplification module (SAM). (b) CBAM, consisting of channel attention and spatial attention modules. (c) Squeeze and excitation block (SEB).

By adding these attention weights to the initial feature map, ability to either amplify or suppress the channel-wise information and train the network to focus on the most discriminative channels is possible. On the other hand, the SAM made it simpler to record spatial location interdependencies inside the feature map. The SAM accomplished this by using ReLU activation and two convolutional layers, each with a 7×7 kernel size. These layers returned spatial descriptors that indicated regional geographic trends. To develop spatial attention maps, sigmoid activation function is used to the spatial descriptors that were created. The resulting attention maps were then element-wise multiplied with the original feature map. This method highlights important geographic areas while suppressing superfluous ones.

Synchronous Amplification Module (SAM): To encode salient spatial and channel information concurrently, the convolution layer captures local spatial features across all channels, as shown in Fig. 3 Left. While bringing out the channel-wise discriminative structural component features, we selectively draw attention to and repress other aspects. The SAM analyses critical spatial and channel data separately to resolve these issues and improve performance. SAM compresses the spatial plane of the input tensor using global average pooling before stimulating it channel by channel to obtain the channel information. The module can automatically include the global channel description due to the channel compressing operation, which provides channel-by-channel statistics for the entire image. The subsequent dense layers use non-linear adaptive re-calibration to extract discriminative channels with significant characteristics, while also utilising contextual channel information. In order to generate the channel attention feature map Ch(I) depicted in Fig. 3 Left, the output of two dense layers is multiplied by the sigmoid function and the input (I). This is done to highlight the characteristics necessary for channel-specific identification. In a manner analogous to how the first portion of the module for simultaneous activation compresses the channels, the second portion of the module employs convolution blocks to capture the spatial characteristics shared by all channels. The recovered features are spatially stimulated and the output is then multiplied by the input tensor to emphasise the essential spatial data Sp(I). In contrast to where spatial attention is conducted via an average and maximum pooling operation, the global channel features are compressed to extract salient spatial information for the provision of spatial statistics by reducing the input through the channel dimension. Instead of employing 1×1 convolution directly for the aggregation of spatial information, an additional 3×3 convolution block is placed before it to aid in the extraction of features. In this instance, in addition to spatial, Sp(I), and channel, Ch(I), information, the input is also added using a skip connection to prevent the loss of critical discriminative information and to alleviate the vanishing gradient problems [1], as shown in Fig. 3 Left. Together, CAM and SAM improved CBAM's ability to learn and discriminate between more usable representations.

2.2 Aggregated Multi-Feature Attention Module

We developed a group of attention modules with multiple features to aggregate various representations of the local parallel feature extraction process and encode significant data from visually identical concrete structural components, as shown in Fig. 3 Right. The purpose of this module is to include highly localised feature selection mechanisms to differentiate non-bridge components, columns, beams and slabs, other structural and non-structural components. The proposed attention actions in AMFA are executed multiple times to ensure the selection of the most important aspects. Each attention module utilises three dense layers to perform concurrent computations for producing parallel non-linear projections in feature space. The input, I, and its height, breadth, and number of channels are considered. Then, the outputs, T2 and T3, are elementally multiplied, a SoftMax function is applied to produce the attention mask, and T1 is multiplied with the attention mask to emphasise the most critical features. The AMFA module produces its output by merging the attentive features from various representations generated by three different attention operations, namely, att1, att2, and att3.

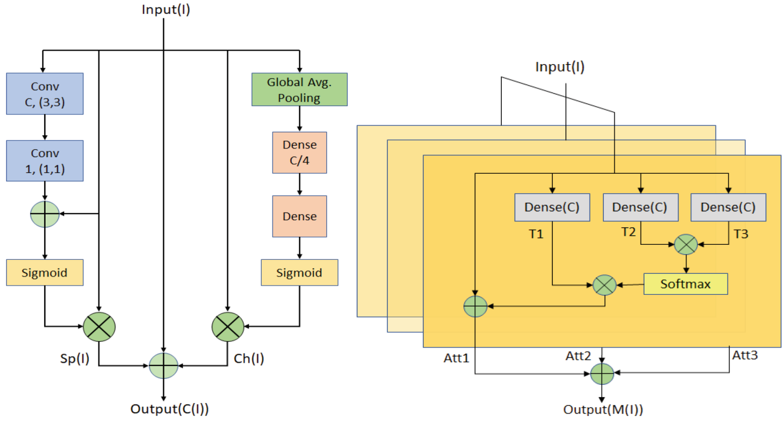


Fig. 3. Left: Synchronous amplification module (SAM) consist of global average pooling for squeezing, followed by dense layers to extract contextual channel features. The other module applies convolution layers to the input to capture spatial features across all channels. Right: Aggregated multi-feature attention module (AMFA) consists of 3 attention layers, each attention layer processes the input simultaneously. The input I is passed through a dense layers. The resultant outputs T2 and T3 are multiplied element-by-element, and a softmax function is then applied to generate an attention mask. The generated mask is then multiplied by T1 in order to emphasise key features. In order to accomplish identity mapping, the output is appended to the input tensor.

2.3 Squeeze and Excitation Block

Instead of recording positional or global dependencies, the squeeze and excitation block (SEB) module adapted from [9], focuses on recalibrating the relative relevance of specific channels inside a feature map, as shown in Fig. 1 and details in Fig. 2(c). By emphasising informative channels while suppressing less important ones, it strengthens the network's ability to discriminate. The squeeze layer and the excitation layer are the two main layers that make up the SEB module. The squeeze layer applies global average pooling over the spatial dimensions of the input feature maps to produce spatial compression. The process captures channel-wise statistics and aggregates data across the spatial dimensions by reducing each channel to a single value. The excitation layer models and captures the interactions and interdependencies between channels. The excitation squeeze layer and the excitation excitation layer are its two sub-layers. Excitation squeeze layer utilises a fully linked layer with a lot fewer units than the initial number of channels, the excitation squeeze layer further compresses the channel-wise information. This compression preserves crucial channel features while assisting in the reduction of computing complexity. The excitation layer restores the original number of channels to the compressed descriptor. It uses an activation function like sigmoid or ReLU, followed by a second fully linked layer. This layer produces channel-specific attention weights that show how crucial each channel is for capturing differentiating elements.

3 Experimental Results, Analysis and Discussions

In this work, at first we study the impact of attention mechanism on the interpretability and performance of image classification models on the bridge component classification dataset [13,14]. The experiment involved utilisation of explainable AI visualisations techniques (XAI) [17] on both traditional classification models with/without attention mechanisms and compare them with our IAN architecture.

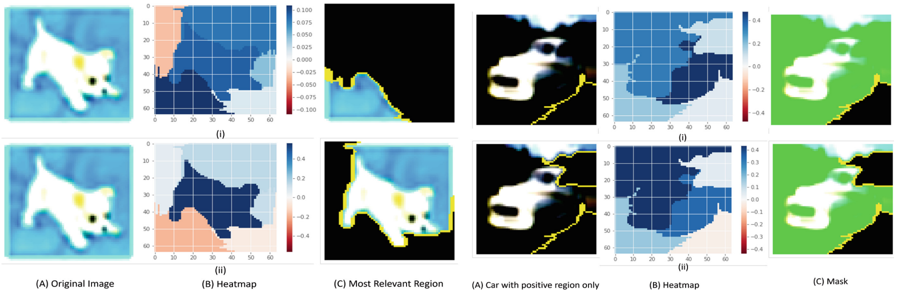
3.1 Bridge Component Classification Dataset

Bridge component classification (BCC) dataset is provided by the authors of [13,14]. This dataset consists of 1,563 photographs of bridges acquired for research and comparative evaluation purposes. Using their provided partitions, 1329 images are used for training, while the remaining 234 images are utilised for testing. Each image is labelled into five categories: Non-bridge, Columns, Beams and Slabs, Other Structural and Non- structural components. The dataset comprises images with dimensions of 320 × 320 pixels. We used this dataset to train our proposed model and compared with the existing methods to check its performance.

3.2 XAI Visualisations

In this study, we aim to compare the interpretability and performance of the models with and without attention mechanism and identify any differences in their behaviour. To enable attention mechanism, the models are modified by including attention layers that helped them focus on important features of the input image. Popular XAI techniques such as local interpretable model-agnostic explanations (LIME) [19] is used to provide insights into how the models make their predictions, enabling the development of more trustworthy and interpretable image classification models. At first, we conducted our experiments on ResNet18 [8], which is a popular deep residual neural network models widely used for image classification. The experiment is carried out on the CIFAR-10 dataset [11], a standard dataset for image classification consisting of 60,000 images divided into 10 classes, each with 6,000 images.

Visualisation on ResNet18: We did experimental analysis of ResNet18 model with and without the inclusion of attention module, using the CIFAR-10 dataset [11] for a classification task. Specifically, we evaluate CBAM attention module to assess its impact on the model's performance. In this context, Fig. 4 shows the LIME visualisation outputs for two selected images, which serve as an important step towards evaluating the model's interpretability and ensuring its explainability. On left top in Fig. 4(i), the image depicts a 'dog' that is accurately predicted by the model. However, to determine which region or cluster have the most significant influence on the model's prediction, LIME XAI is



(A) Original Image (B) Heatmap (C) Most Relevant Region (A) Car with positive region only (B) Heatmap (C) Mask

Fig. 4. For ResNet18 model [8] on CIFAR-10 dataset [11], left top row, in (i) without attention module to predict image as a 'dog' but the most important or dominant region highlighted by LIME is not accurate, but in left bottom row, (ii) when CBAM attention is employed to the model, focus of model is improved and LIME visualised more accurate region as the dominant for predicting it as a dog. Right top row, (i) Heatmap shows less important region to a human as more important for the model's prediction as a 'car', However when attention is employed as in right bottom row, (ii), region which is more relevant to a human is also more important for the model according to the LIME visualisation. For both center images ((B)Heatmap), dark blue region represents more importance of the region to model for positive prediction.

utilised to replicate the model. The resulting visualisation is shown in Fig. 4(i) 'Most relevant region'. The region displayed in the visualisation is less relevant. The heatmap in Fig. 4 (i) provides an observation of all the important regions for the model by using the intensity of blue color. As the superpixel becomes dark blue (or more blue), it indicates that it is more critical for the model's prediction whereas the red region represent the superpixel which impacted negatively for the prediction of a model. When an attention module (CBAM in this case) was added to the model, however, the visualisation improved and the region identified by the LIME XAI technique became more accurate which can be seen in Fig. 4(ii), or we can say bias in the model is reduced. Similar observation can be drawn with the second image of a 'car' in Fig. 4 right, top and bottom.

Visualisation Using our Proposed IAN: Figure 5 shows the mask generated by LIME using IAN, it depicts the region where the model is focused when predicting for the beams/slabs class on an image. From this Fig. 5 it can be illustrated that the region where model is focusing is very relevant based on the ground truth image. The accuracy of the masked region is compromised because the segmentation is not finely tuned and relies on the chosen segmentation algorithm. When the algorithm forms clusters that encompass both positive and negative regions, the resultant output includes the entire cluster, leading to a reduction in the accuracy of the visualisation. There are few points that can be conjectured based on observation pertaining to this use case. Firstly, LIME's effectiveness is influenced by multiple factors beyond just emulating the model. The approach employs different segmentation algorithms to partition images into clusters or superpixels. However, if a single pixel in the region is active, the entire cluster generated during segmentation is taken into consideration sometimes, which diminishes the accuracy and performance of the model, which is an issue if the images has noises in the dataset, which is currently the case with the dataset we are using, detail analysis can be found here [10,21].

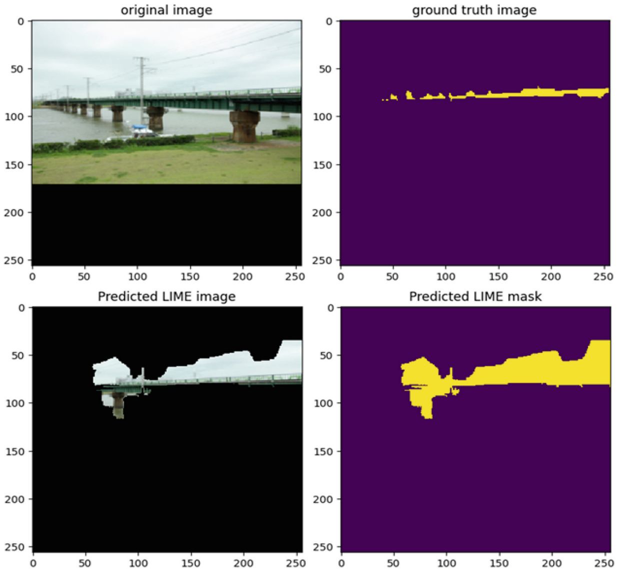


Fig. 5. IAN's LIME visualisation highlights the key features within the image that hold significance for the model's prediction of the beams/slab class on the bridge component dataset [13,14].

Secondly, the accuracy of surrogate models is influenced by the number of perturbations used during training [12]. Fewer number of superpixel combinations can negatively impact the model's subsequent predictions and sometimes lead to the prediction of clusters that are not part of the ground truth. Lastly, various segmentation algorithms, such as SLIC [16], Quickshift [23], and Felzenszwalb [6], generate varying numbers of superpixels. The greater the number of superpixels, the greater the number of possible cluster combinations, resulting in finer granularity, and subsequently, improving the performance of the surrogate model. Here granularity refers to the level of detail or resolution of something. In the context of image segmentation, granularity refers to the level of detail in the segmentation or clustering of pixels into superpixels. A higher granularity indicates a finer segmentation with more distinct superpixels, whereas a lower granularity indicates a coarser segmentation with fewer but larger superpixels.

3.3 Implementation Details

In our proposed IAN model we employ max pooling operation which helps in retaining important features, suppresses noise, and promotes generalisation [15]. For training process, the batch size used is 4. The dataset has been subjected to random cropping, random flipping, and random rotation in addition to the centre crop, in accordance with the previous report works [10,14,21]. Binary cross entropy loss function is employed for learning of model. The learning rate is set at 0.001 at this time. Adam optimiser is used for smooth weights update and training process of model with Beta1 = 0.9 and Beta2 = 0.999. Number of epochs used to train this model on the BCC dateset is 100. Experiments are conducted on a machine equipped with an Intel Xeon W-2123 CPU with 3.60 GHz, 96 GB of RAM, and an NVIDIA Titan XP 8 GB GPU card using the Python Keras API and TensorFlow backend.

3.4 Performance Metrics

We employed two metrics: pixel accuracy (PA) and mean of intersection over union (mIoU) to evaluate the performance of our model and conduct comparisons with other benchmark models, similar to [10,21]. It assesses the model's effectiveness by quantifying the number of pixels accurately identified by the model in relation to the total number of pixels. However, this metric has a limitation in that it does not provide a comprehensive evaluation of the model's segmentation performance. Details are provided in [10,21], with examples of visualisation Fig. 5 and explanations in Section 3.3 in [21]. An effective metric for assessing a segmentation model's performance is the intersection over union (IoU). IoU provides a more thorough and insightful assessment of the segmentation model's performance in comparison to metrics like pixel accuracy, which only count the proportion of properly categorised pixels. It also considers sensitivity to tiny items. For instance, the mean of IoU (mIoU) of the input image would be much lower compared to pixel accuracy when only two classes (background and structural component) are taken into account, showing the larger relevance of IoU in evaluating segmentation performance, details are here [21].

3.5 Comparison with Benchmarks

Table 1 summarises the performance of the proposed architecture over the other benchmarking techniques [10,13,14,21,25]. Most of the other results are different approaches proposed by [13], which uses three architectures FCN45, SegNet45 and SegNet-S, using three different configurations (naive configuration, parallel Configuration and sequential configuration). Both mIoU metric and pixel accuracy (PA) metrics are used to compare the performance of his most recent suggested naive (N), parallel (P) and sequential (S) models with different configurations. Out of all the methods by [13], FCN45-N evidenced to be the most successful, with a pixel accuracy of 84.1% and a mIoU of 57.0%. [10,21] most recent research on this perform better than earlier models. In contrast to [21]

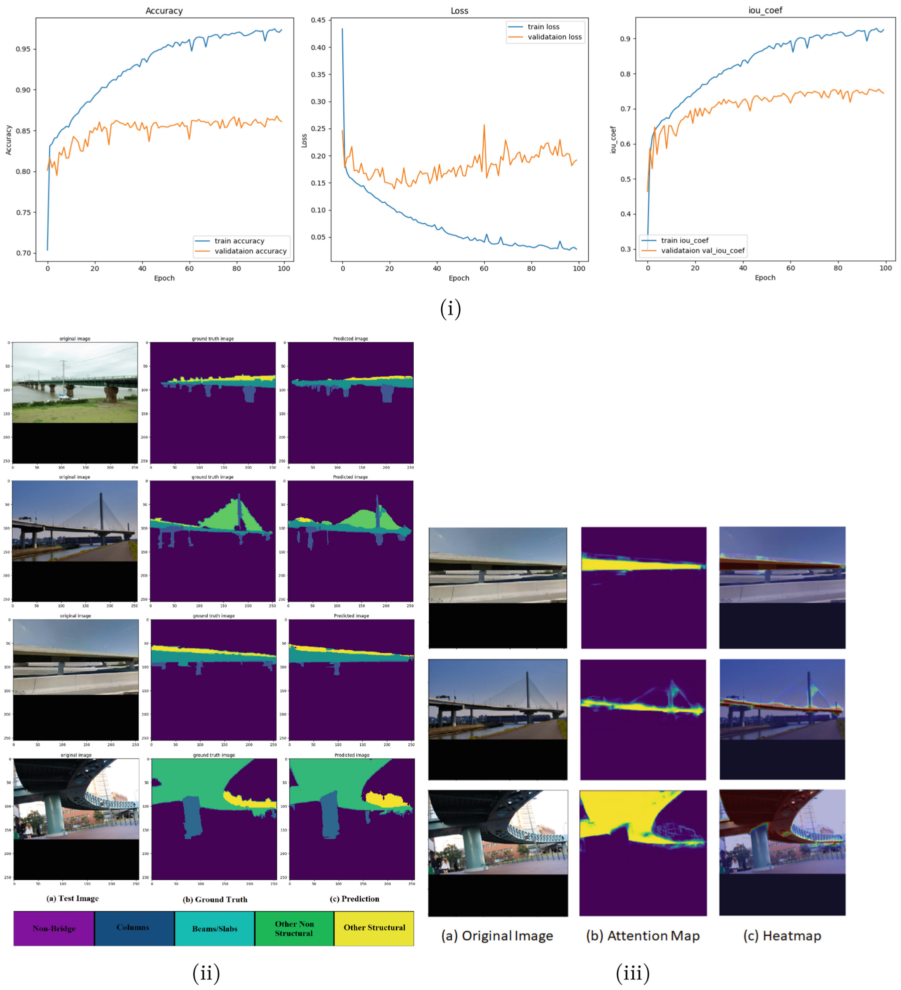
Table 1. Comparison with benchmarks evaluating intersection over union (mIoU) and pixel accuracy (PA), for comparison of different models on bridge component classification dataset [13, 14]

Benchmarking Methods	mIoU(%)	PA(%)
CNPT-N [14]	50.8	80.3
CNPT-Scene [14]	–	82.4
FCN45 [13]	–	82.3
FCN45-N [25]	57.0	84.1
FCN45-P [25]	56.9	84.1
FCN45-S [25]	56.6	83.9
SegNet45-S [25]	54.5	82.3
SegNet45-N [25]	55.2	82.9
SegNet45-P [25]	55.2	82.9
SegNet45-S-N [25]	55.8	83.1
SegNet45-S-P [25]	55.9	83.3
SegNet45-S-S [25]	55.4	82.7
StructureNet [10]	57.46	89.08
DNNAM [21]	65.94	82.85
UNet [20]	67.8	83.21
IAN	**71.02**	**85.09**

model, which has a pixel accuracy of 82.85% but a mIoU of 65.94%, which is higher than the StructureNet model [10], proposed by [10] has the highest pixel accuracy of 89.08% and a mIoU of 57.46%. Our proposed IAN model outperforms all other models with the highest mIoU of 71.02% and the second-highest pixel accuracy of 85.09%. Since mIoU offers a class-specific evaluation and handles class imbalance, providing a more detailed, balanced and effective assessment of segmentation models compared to pixel accuracy, our emphasis has been directed towards prioritising mIoU as the primary evaluation metric. Hence, in terms of mIoU, our proposed IAN model surpasses the previously established highest benchmarks by 5.08%, demonstrating superior performance.

The convergence curves during the training process are shown in Fig. 6(i), which demonstrate how well our proposed model performs, with loss continuously decreasing and accuracy and IoU continuously rising. However, it should be noted that the performance of models using validation data gets saturated due to the irregular labelling of few ground truth images on the BCC dataset. IAN's average processing time for a 256 × 256 input image is 0.081 s.

Figure 6(ii) shows the output generated by the proposed IAN model on few images from BCC dataset. Figure 6(iii) shows the activation maps followed by heatmaps, which helps in depicting how attention mechanism are helping the model to improve the focus more on correct component or region in order to correctly distinguish it in an image. Furthermore AMFA, SEB and PTAM attentions employed in our model combined helps in capturing long-range dependencies and integrating information from distant regions in the feature maps to improve the performance and understanding of the model.



(i)

(a) Test Image (b) Ground Truth (c) Prediction

(ii)

(a) Original Image (b) Attention Map (c) Heatmap

(iii)

Fig. 6. (i) Training process curves showcasing model performance over iterations on different parameters. Blue represents the training set, while orange represents the validation set. (ii) Segmentation results from our proposed IAN model on images of Bridge Component Classification Dataset. They highlight that employing attention mechanism has significant affect on the outputs. (iii) Attention maps and heatmaps obtained by the proposed model for different images on Bridge Component Classification dataset. They highlight the regions that are dominant for the prediction of a particular class.

4 Ablation Studies

To investigate the individual contributions and effects of various components on the performance of our model, ablation study is conducted. By selectively removing or altering specific modules, we aim to assess their impact on the model's performance. Initially, we established a baseline model (adapted UNet [20]) that represents the network architecture without any attention modules. Subsequently, we systematically removed or modified specific components, while keeping the remaining system intact. The performance of each modified configuration is then evaluated and compared against the baseline network. At first, we omitted the PTAM placed at the core of network, and SEB attention module is placed at the core of network, a drop in the performance is observed, as noted in Table 2.

Table 2. Ablation Study on the attention mechanism employed by proposed IAN Model on BCC dataset [14]. mIoU and Pixel Accuracy metric is used to check the performance of different configurations

Models	mIoU(%)	PA(%)
UNet [20]	67.8	83.21
Baseline architecture (adapted UNet [20])	69.5	84
only SEB	70.3	83.9
only PTAM	70.3	84.2
SEB + PTAM	70.5	84.1
PTAM + AMFA	70.7	84.5
PTAM at internal 2 layer + SEB at center	70.2	84.02
SEB at 2 outer layer + PTAM at center	70.6	84.3
SEB+ AMFA at all layer + PTAM at center	69.8	83.9
PTAM with SAM only	69.97	84.39
PTAM with CBAM only	70.27	84.80
IAN	**71.07**	**85.09**

Without PTAM module, model struggle to effectively capture channel-wise dependencies in feature maps and spatial wise dependencies within a feature map and assign appropriate weights to different channels. This can lead to suboptimal feature extraction, reducing the discriminative power of the model and hindering accurate segmentation. It can also result in difficulties in capturing spatial relationships and attending to critical spatial locations. This may lead to challenges in accurately delineating object boundaries, handling occlusions, and capturing fine-grained details. The model's segmentation performance may suffer as it fails to focus on relevant spatial regions and adequately suppress irrelevant or noisy parts of the input.

Similarly, performance drop is seen, when the SEB module is omitted and only PTAM is placed at the core, due to the lack of spatial attention to the input image at outer layers, it becomes difficult to focus on individual objects and struggle to prioritise informative regions in the image. Subsequently, we investigated the insertion of these attention mechanisms between various intermediate layers of the model architecture and evaluated the resulting performance which can be seen in Table 2. The findings indicate that when these attention mechanisms are placed in accordance with the proposed methodology, they produce the most beneficial and optimal results.

Next, we tested the efficacy of AMFA module with PTAM module. First we disabled the SEB module, only AMFA was enabled at the inner two layer, then we enabled the SEB and disabled AMFA. It can be noted from the observed results in Table 2 that each module does not provide optimum results individually, but when both enabled simultaneously significant improvement can be observed. Disabling AMFA causes lack in highly localised feature selection and may struggle to capture intricate boundaries, handle overlapping objects during segmentation. Later on we tried with different configuration of these modules, experimental results are noted in Table 2.

5 Conclusions

In this work, an interpretative attention network (IAN) is proposed for handling the challenges involved in structural component recognition. It incorporates attention mechanism that enables the model to focus on relevant regions automatically, help in efficient training and resulted in improved recognition accuracy. The proposed architecture incorporates innovative parallel twin attention module, synchronous amplification module, aggregated multi-feature attention module and squeeze and excitation blocks, that helps to capture long-range dependencies, allowing the model to take contextual information into consideration when segmenting objects. We utilised XAI techniques such as LIME to visualise the efficacy of the attention mechanism. We illustrated how the model allocated its attention and the effectiveness of the attention mechanisms in capturing pertinent image regions. In order to assess the performance of the proposed model, various evaluation metrics, including pixel accuracy and mIoU are utilised. Additionally, ablation study is conducted to further examine the model's performance by systematically analysing the effects of removing specific components or network modules. Experimental results on various benchmarking datasets demonstrate the superiority of the proposed architecture over the existing approaches.

References

1. Basodi, S., et al.: Gradient amplification: an efficient way to train deep neural networks. Big Data Min. Anal. **3**, 196 (2020)
2. Bhattacharya, G., Mandal, B., Puhan, N.B.: Interleaved deep artifacts-aware attention mechanism for concrete structural defect classification. IEEE Trans. Image Process. **30**, 6957–6969 (2021)

3. Bhattacharya, G., Puhan, N.B., Mandal, B.: Kernelized dynamic convolution routing in spatial and channel interaction for attentive concrete defect recognition. Signal Process. Image Commun. **108**, 116818 (2022)
4. Bhattacharya, G., Puhan, N.B., Mandal, B.: Stand-alone composite attention network for concrete structural defect classification. IEEE Trans. Artif. Intell. **3**(2), 265–274 (2022)
5. Farabet, C., Couprie, C., et al.: Learning hierarchical features for scene labeling. IEEE PAMI **35**(8), 1915–1929 (2013)
6. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. Int. J. Comput. Vis. (2004)
7. Gallagher, R.P.: Earthquake Aftershocks—Entering Damaged Buildings, Applied Technology Council (1999). https://www.atcouncil.org/pdfs/atc35tb2.pdf
8. He, K., Zhang, X., et al.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
9. Hu, J., Shen, L., Albanie, S., et al.: Squeeze-and-Excitation Networks (2019)
10. Kaothalkar, A., Mandal, B., Puhan, N.B.: Structurenet: deep context attention learning for structural component recognition. In: 17^{th} International Conference on Computer Vision Theory and Applications (VISAPP), pp. 567–573 (2022)
11. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images (2009). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
12. Mishra, S.S., Mandal, B., Puhan, N.B.: Perturbed composite attention model for macular optical coherence tomography image classification. IEEE Trans. Artif. Intell. **3**(4), 625–635 (2022)
13. Narazaki, Y., et al.: Vision-based automated bridge component recognition with high-level scene consistency. Comput. Aided Civ. Infrastruct. Eng. **35**(5), 465–482 (2020)
14. Narazaki, Y., Hoskere, V., et al.: Vision-based automated bridge component recognition integrated with high-level scene understanding (2018)
15. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV) (2015)
16. Radhakrishna, A., et al.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE PAMI **34**(11), 2274–2282 (2012)
17. Ras, G., et al.: Explainable deep learning: a field guide for the uninitiated (2021)
18. Register, F.: Federal highway admin, department of transportation (2004). https://www.govinfo.gov/content/pkg/FR-2004-12-14/pdf/04-27355.pdf
19. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": explaining the predictions of any classifier (2016)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
21. Sarangi, S., Mandal, B.: Deep neural network based attention model for structural component recognition. In: 18^{th} International Conference on Computer Vision Theory and Applications (VISAPP), pp. 317–326 (2023)
22. Spencer, B.F., Hoskere, V., Narazaki, Y.: Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering **5**(2), 199–222 (2019)
23. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: European Conference on Computer Vision (ECCV), pp. 705–718 (2008)
24. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional block attention module (2018)
25. Yeum, C., Choi, J., Dyke, S.: Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure. Struct. Health Monit. **18**, 675 (2018)

Pedestrian Attribute Recognition Using Hierarchical Transformers

Lalit Lohani[1], Kamalakar Vijay Thakare[2(✉)], Kamakshya Prasad Nayak[1], Debi Prosad Dogra[1], Heeseung Choi[2,3], Hyungjoo Jung[2], and Ig-Jae Kim[2]

[1] Indian Institute of Technology (IIT), Bhubaneswar 752050, India
{22cs06009,a22ee09018,dpdogra}@iitbbs.ac.in

[2] Artificial Intelligence and Robotics Institute, Korea Institute of Science and Technology, Seoul 02792, Republic of Korea
{drkam,jhj0220,drjay}@kist.re.kr

[3] Yonsei-KIST Convergence Research Institute, Yonsei University, Seoul 03722, Republic of Korea
hschoi@kist.re.kr

Abstract. The goal of pedestrian attribute recognition (PAR) is to detect and classify a wide range of pedestrian attributes, such as gender, carrying objects, clothing styles, body postures, age groups, and more. It plays a vital role in computer vision, specifically in crucial applications such as behaviour analysis, public safety monitoring, and video surveillance. However, existing PAR approaches are unable to achieve substantial performance due to multiple factors. First, multiple appearances of the same attribute confuse the models. Second, adverse weather and lighting conditions restrict model generalization capability. To mitigate these challenges, this paper proposes a new evaluation baseline that uses Vision Transformer (ViT) blocks for hierarchical feature modelling. The approach categorizes attributes into different spatial granularity levels and employs diverse patch formations to extract discriminative features. Furthermore, we introduce an enhanced loss function for stable training in the re-formulated granularity scenario, where a novel attribute-aware granularity factor influences the loss. The proposed baseline has been extensively evaluated on the three popular PAR datasets, namely RAP, PA100K and PETA.

Keywords: Pedestrian Analysis · Vision Transformer · Video Surveillance · Attribute Recognition

1 Introduction

The task of Pedestrian Attribute Recognition (PAR) attempts to identify a wide spectrum of attributes such as *gender*, *age group*, *profession*, *body shape*, *posture*, *hair structure* or *clothing styles* of pedestrians. In recent years, PAR has achieved large interest due to its critical application in public safety and surveillance. This significant enhancement has initiated the implementation of

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 78–93, 2025.
https://doi.org/10.1007/978-3-031-78444-6_6

few important use-cases such as person re-identification [6,7,29] and scene analysis [28]. In addition to this, few researchers have utilized advanced deep learning based techniques to enhance the model's recognition performance. Tang et al. [23] employed feature pyramid architecture to train end-to-end network. A pioneer CNN-based model proposed by Li et al. [12] utilize multi-attribute training to train the baseline model. Jia et al. [10] proposed disentangled attribute feature learning (DAFL) framework to learn disentangled features from the input images. Tang et al. [24] combined multiple pedestrian datasets and trained the common transformer network to achieve substantial performances on downstream tasks. Thakare et al. [25] attempt to predict a wide range of attributes from multi-view input setting. On the other hand, approaches proposed in [8,31] employ *ResNet-50* as backbone and extract high-level features from single image and learn semantic consistency across the attributes.

Pedestrian Attribute Recognition (PAR) also gained popularity due to the availability of a few benchmark datasets. The popular PAR datasets are PETA [3], RAP [14], Market1501 [16] and PA100K [18] include several important attributes such as *gender*, *cloth patterns*, *accessories*, *viewpoints*, etc. However, the generalization ability of the trained models is restricted by the data imbalance in these datasets. For example, the Market1501 dataset consists primarily of 94.4% of images featuring individuals wearing *short sleeves*. Furthermore, inaccurate annotation guidelines pose challenges; for instance, in the PETA dataset, the age attribute spans a wide range, yet it is assigned as age 18-60, encompassing over 73% of samples. However, a handful of recent PAR approaches have correctly addressed *data imbalance* problem. Specker et al. [20] introduced UPAR dataset which is a unified version of RAP [14], PETA, PA100K, and Market1501. Li et al. [12] and another recent study [21] obtained attribute positive ratio and integrated it with cross-entropy loss to increase the recognition capability of the model.

Though the performance of recent PAR approaches is commendable, their robustness and generalization are constrained by three major problems. Firstly, multiple appearance of the same attribute hampers the accuracy of the prediction. For instance, in a crowded scene for the Person Re-Identification task mentioned in these studies [6,7,29], many pedestrians wear identical hats. The similarity in appearance makes it difficult to identify individuals solely based on their appearance. Jia et al. [8] have correctly identified this similarity problem and proposed semantic attribute consistency module. Secondly, the varying weather and lighting conditions alter the visual clues of the attributes. For example, it is difficult to predict the presence of the *"black hair style"* in the dark. Lastly, data imbalance within the PAR datasets hampers the generalization and robustness of PAR models. To address these challenges, the proposed solution is expected to exhibit in-variance to complex weather conditions and similarities in features.

To address the above complexities, an effective approach is to classify attributes into distinct categories based on their appearance patterns. For example, the appearance of *gender* aggregates information from various attributes such as *hair pattern*, *clothing style*, and *accessories*. Jia et al. [8] categorize attributes based on their semantic consistency; for instance, items like *helmet*, *hood*, and *bucket hat* are grouped together within the same semantic embedding space as *hats*. Similarly, the authors of ParFormer [5] investigate the interrelationships between groups of attributes in conjunction with the pedestrian viewpoint in the given image. Inspired by this approaches, we provide a strong baseline for hierarchical feature modelling by employing Vision Transformer (ViT), in which the attributes are divided and recognized into 3 spatial granularity levels. Concerning hierarchical feature modelling, we first grouped the set of attributes according to their spatial distribution and granularity levels. We then extract discriminative features using ViT blocks. Attributes like *age*, *gender*, and *viewpoint* lack spatial region and necessitate global-level feature analysis representing low-level granularity. Conversely, attributes like *face masks*, *boots*, and *glasses* can be spatially located in an image, enabling their recognition through attention to a specific region. Therefore, based on their spatial granularity characteristics, we have categorized the attributes into 3 levels and utilized diverse patch formations prior to extracting shared features using ViT blocks. We have explained these levels in Sect. 3.2.

ViT [4], which has been utilized for various computer vision tasks due to its ability to learn discriminative features through multi-head self attention, has been used in our baseline. The recent work proposed in [1] and ParFormer [5] have employed variants of popular Transformer networks. However, they significantly differ from proposed approach. The study proposed in [1] consists of two pre-processing stages: (i) it employs *ResNet-50* as backbone and extracts high-level features from single image into three different feature map formats $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}_3$. (ii) Then each feature map undergoes another sub-layer, where authors have utilized attention masks and series of convolution layers to extract attribute-wise features. On the other hand, our approach first divides the single image (unlike [1]) into patches according to the granularity level and attribute-wise features have been extracted by series of Vision Transformers blocks. The approach mentioned in [8] also employs *ResNet-50* to extract high-level features and it obtains spatial attention maps to segregate positive-negative samples for better training. Their proposed Semantic Consistency Module takes care of obtained attribute-wise features, which is significantly different than our proposed approach. More technical discussion about our architectural choice can be found in Sect. 3.3. In addition to the feature extraction framework, we have also introduced an enhanced loss function tailored for effective model training in the proposed granularity scenario. Our experiments reveal that, higher granularity demands more discriminative features for precise detection. Moreover, we have integrated a logical combination of positive ratio and granularity into the loss function for stable training.

Putting all this together, the paper makes the following contributions: i) We attempt to address Pedestrian Attribute (PAR) task by formulating granularity setup, where feature extraction and learning take place based on attribute-wise granularity levels through a series of Vision Transformer blocks. ii) We also address the inherent data imbalance problem in PAR via proposing a granularity-based penalty in training loss. iii) To demonstrate the effectiveness of the proposed setup and loss, we have conducted a wide range of experiments on the popular PAR benchmark datasets.

2 Related Work

In recent years, Pedestrian Attribute Recognition (PAR) has attracted interest due to its applications in person retrieval, re-identification, and behavior understanding. This section examines a number of notable efforts that have contributed to the PAR.

In the past few years, deep learning-based architectures such as CNN and LSTM have been popular choices for mainstream computer vision tasks. Several PAR approaches have also utilized these architectures to mitigate the complexity exists in PAR. A handful of PAR methods are CNN-based [13,17] or utilized time-series-model [15]. Cheng et al. [2] introduced a visual-textual pipeline for Pedestrian Attribute Recognition (PAR), treating it as a multi-modal task to leverage intrinsic textual information within the attribute annotations. Li et al. [15] employed a continual learning approach to manage multiple groups of pedestrian attributes, integrating a self-learning method to address inconsistent labels. Jia et al. [10] have highlighted the constraints of the one-shared-feature-for-multiple-attributes approach, opting instead for a disentangled attribute feature learning framework. Thakare et al. [25] and Fan et al. [5] leverage relationship between attributes with respect to different viewpoints. Specifically, authors of [5] have proposed Multi-view Contrastive Loss (MVCL) to exploit viewpoint information into network, whereas approach mentioned in [25] extract class activation information from all available viewpoints and fuse it to obtain robust prediction. Bui et al. [1] extract global-level features using a combination of both Swin and Vision Transformer and fused it using cross fusion technique.

In addition to these approaches, several frameworks centered around classifiers address the inherent data imbalance in PAR. In data-centric framework, the minority class augmented through oversampling, or the majority class can be scaled down to ensure stable training. On the other hand, loss-centric approaches involve training the classifier with novel loss functions that account for various scenarios, such as viewpoint and the positive ratio of attributes, among others. One of the pioneer work in PAR [12] introduced the weighted binary cross-entropy loss function to effectively address data imbalance. Jia et al. [10] additionally suggested employing a triplet loss to aid the group attention merging module in learning discriminative features. Yan et al. [30] proposed incorporating a dropping rate during training, coupled with delaying the training of hard samples, thereby favoring easier samples.

Though the recent works employ Transformer-based methods detailed in [1, 2,5], however, it differs in several critical aspects: (i) Unlike the unified patch embedding utilized in [1,5], our method adopts a granularity-based patching technique. (ii) By integrating hierarchical patching at the preprocessing stage, our framework eliminates the necessity for hierarchical feature extraction models. (iii) Contrary to the viewpoint-centric loss applied in [5], we introduce a novel granularity-based loss function for training. In the next section, we provide a details description of the proposed method.

3 Proposed Baseline Method

PAR models expect to identify and characterize pedestrian attributes across a wide spectrum of vastly differing conditions and environments. We propose a strong evaluation baseline for recognizing attributes. The high-level architecture of the baseline model is depicted in Fig. 1. It consists of three stages: (i) Concerning hierarchical feature modelling, we have grouped attribute sets into 3 levels according to their spatial distribution. Patch formation and positional embedding are done as per the three granularity level. (ii) In the second stage, we employ multiple Vision Transformer (ViT) blocks to extract features from embedded patches at each level. (iii) Lastly, a level-wise MLP-Head is trained with custom loss to predict the probability for level-wise attributes. In the next subsection, we discuss each stage in detail.

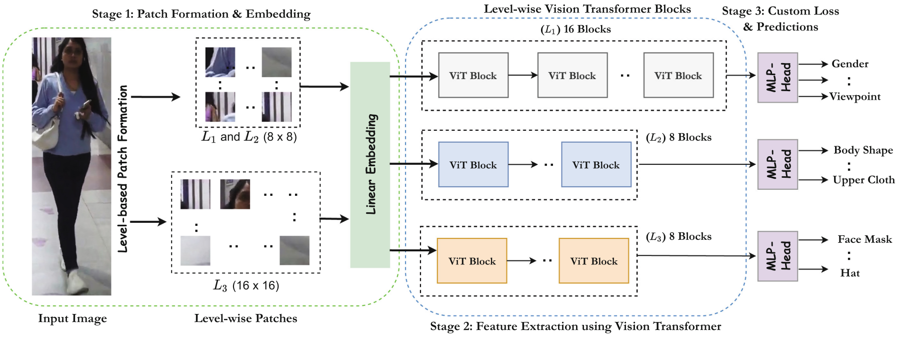


Fig. 1. Proposed Strong Baseline: The proposed baseline comprises of three stages: (i) According to the spatial granularity level, the input image is divided into non-overlapping patches, and positional embedding is added. (ii) Embedded patches are then fed to the corresponding series of ViT blocks for feature extraction. (iii) Finally, MLP-Head of each block is trained with the proposed loss function and the probability for level-wise attributes, is predicted.

3.1 Problem Formulation

Following the prior works [2,12,30], we formulate the PAR problem as a multi-label classification problem, where the model expects to learn discriminating features that represent the presence or absence of attributes in a given pedestrian image.

Assume attribute set of a pedestrian image is denoted by $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}$, where K is the number of attributes. Let $\{(\mathcal{I}_1, \mathcal{Y}_1), (\mathcal{I}_2, \mathcal{Y}_2), \ldots (\mathcal{I}_N, \mathcal{Y}_N)\}$ be N image samples in the training set, where $\mathcal{I}_i$ is the i-th pedestrian image and $\mathcal{Y}_i \in \Pi$. More precisely, $\mathcal{Y}$ represents a human-annotated binary vector wherein 0 and 1 denote the absence and presence of an attribute in the image $\mathcal{I}$, respectively. In this scenario, our aim is to train a Pedestrian Attribute Recognition (PAR) model denoted as $\mathcal{H}(.)$, which calculates the probability p_i for each attribute π_i within the set Π, represented as $\mathcal{H}(\mathcal{I}, \Pi) = [p_1, p_2, \ldots, p_M]$. These probabilities (p) are utilized to calculate the loss during training and generate predictive outcomes during inference.

3.2 Spatial Granularity and Embedding

The attribute-wise granularity levels are determined based on the image portion they occupy to be correctly detected. For example, the number of pixels required to detect attributes like *hair color* or *boot color* is significantly smaller as compared to attributes such as *gender*, *age*, or *viewpoint*. To establish these levels, we have divided pedestrian images from the PAR dataset into multiple patches and analyzed the level of appearances. This analysis has involved reviewing hundreds of images by multiple annotators, leading to the formation of three distinct granularity levels. In addition to our observations, we have employed object detection models like Faster R-CNN and YOLO-V3 to detect specific attributes such as *face masks*, *boots*, and *bags* from cropped patches. Upon successful detection, it has confirmed the low-level granularity of these attributes through cropped patches from pedestrian images. Therefore, the proposed granularity levels are both empirically derived and supported by object detection model's predictions. Figure 2 depicts a handful of attributes from each level and summarizes the attributes from PA100K dataset with their respective levels. For each granularity level, we first divide the image into patches and map each patch to a high-dimensional representation using linear embedding. Moreover, position embeddings are also sequentially added to this vector to retain the positional encoding. These embeddings are then processed through a series of consecutive ViT blocks.

3.3 ViT Encoder Blocks

A handful of contextual factors influence the appearance of attributes; hence, attentive feature learning is essential in a PAR setting. To achieve this, we employ a series of ViT blocks for each granularity level. Each ViT block is basically a regular Transformer [26] encoder, which consists of two sub-layers: *Multi-head self-attention* and *feed-forward* layers. For instance, for the level L_2, we have processed 8×8 embedded patches through a series of 8 consecutive Transformer blocks before feeding them to the classification head. We chose to employ Vision Transformer (ViT) for three key reasons: (i) ViT is directly derived from the popular Transformer architecture with image patch embedding, making it highly

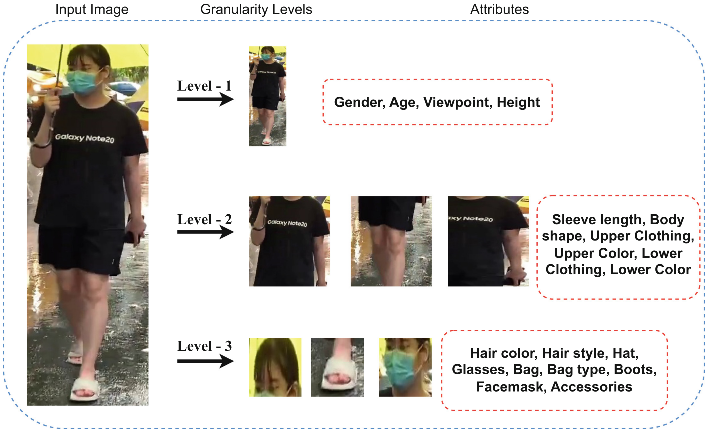

Input Image
Granularity Levels
Attributes
Level - 1
Gender, Age, Viewpoint, Height
Level - 2
Sleeve length, Body shape, Upper Clothing, Upper Color, Lower Clothing, Lower Color
Level - 3
Hair color, Hair style, Hat, Glasses, Bag, Bag type, Boots, Facemask, Accessories

Fig. 2. Spatial Granularity Levels: Three granularity levels, L_1, L_2, and L_3. Abstract attributes such as *gender*, *age* demands full-image analysis for recognition. On other hand, concrete attributes such as *glasses*, *boots* are difficult to localize due to small spatial location.

suitable for handling highly annotated PAR datasets. (ii) ViT allows for faster training and inference due to a lower number of network parameters (86M) and higher throughput (35.9 s per image versus 88M and huge throughput of 120.7 s on PA100K). (iii) Unlike Swin-T, ViT avoids complex hierarchical constructions, which is beneficial since the input consists of predefined hierarchical attribute categories.

3.4 Loss Function

Let $\mathcal{T}_{Train} = \{\mathcal{I}_i, \mathcal{Y}_i\}_{i=1}^{N}$ be the training set, and p_{ij} be the predicted probability by the model $\mathcal{H}$ for $\mathrm{j^{th}}$ attribute of $\mathrm{i^{th}}$ image. In multi-label classification context, the weighted binary cross-entropy loss suggested by [12] is a good choice as the primitive loss function for training the classifier, which is depicted in Eq. 1. Here, y_{ij} is the human-annotated ground truth, and K is the total number of attributes.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\omega_j\left(y_{ij}\log\left(p_{ij}\right) + \left(1-y_{ij}\right)\log\left(1-p_{ij}\right)\right) \tag{1}$$

$$\omega_j = \begin{cases} e^{1-r_j}, & y_{ij}=1 \\ e^{r_j}, & y_{ij}=0 \end{cases} \tag{2}$$

Here, ω_j is the weight factor, and r_j is the positive ratio of the $\mathrm{j^{th}}$ attribute in the set $\mathcal{T}_{Train}$. Several existing PAR works [2,5,12] employ the loss function depicted in Eqs. 1 and 2. Apart from this, a few recent works re-formulate the loss function and incorporate additional aspects such as gradient norm [30] and penalty coefficient [21] to effectively handle data imbalance. We have utilised weighted binary cross-entropy loss and attribute-wise penalty coefficient to handle the imbalanced sample distributions. The updated loss function is depicted in Eqs. 3 and 4.

$$\mathcal{L}_j = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} \mathit{\Omega}_j \left(y_{ij}\log\left(p_{ij}\right) + \left(1-y_{ij}\right)\log\left(1-p_{ij}\right)\right) \tag{3}$$

$$\mathit{\Omega}_j = \begin{cases} \sqrt{\frac{1}{2r_j}}, & y_{ij} = 1 \\ \\ \sqrt{\frac{1}{2(1-r_j)}}, & y_{ij} = 0 \end{cases} \tag{4}$$

The imbalance weight factor, $\mathit{\Omega}_j$ is designed to address class imbalance by assigning greater importance to low-frequency classes. For positive samples ($y_{ij} = 1$), $\mathit{\Omega}_j$ increases when the positive ratio r_j is low, thereby boosting the training loss for such classes. This design ensures that the model emphasizes minority classes, thus balancing the learning process. In addition to this, We integrate an attribute-aware granularity factor, denoted as ζ_j^i, which assists the model in learning more discriminative features at different levels. Our observations indicate that L_1 attributes necessitate global-level feature analysis, and their positive ratio is notably higher compared to other levels. Conversely, for L_3, positive ratios are uneven, thus require more attention during the training. We calculate attribute-aware granularity factor using Eq. 5.

$$\zeta_j^m = [\gamma_m \cdot \frac{1}{\mathit{\Omega}_j}] \tag{5}$$

Here, ζ_j^m is granularity factor for $\mathrm{j^{th}}$ attribute on $\mathrm{m^{th}}$ attribute granularity level. Since, weighted attribute-wise training loss in Eq. 3 necessitates a higher training loss for minority attributes, the new loss favours low granularity level attribute due to special low-level feature analysis requirement. This granularity factor (ζ_j^m) is influenced by two key considerations: (i) the positive ratio of the specific attribute and (ii) the prior assumption that L_1-level attributes occupy a larger portion of visual features due to their appearance, thus requiring less specific attention compared to L_3-level attributes. The $\gamma_m = [0.3, 0.5, 0.8]$ are a set of granularity-base multipliers for $\mathrm{L}_1, \mathrm{L}_2$, and L_3, respectively. These values ensure that the weighted loss is propagated according to the granularity level. For example, the *age* attribute, which belongs to the highest level L1, should receive a small adjustment (0.3) in training loss even with a higher positive ratio r_j. In contrast, the *face mask* attribute, categorized under L_3, should experience an (0.8) times greater fluctuation in training loss due to its complex appearance. The final loss is shown in Eq. 6. Here, L_j is the weighted binary cross entropy loss defined in Eq. 3 and λ being the constant to control the influence of the granularity term.

$$\mathcal{L}_F = \mathcal{L}_j + \lambda \cdot \zeta_j^m \tag{6}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

The PA-100K dataset [18] consists of 100K pedestrian images taken in 598 outdoor scenes, with each image annotated for 26 commonly used attributes. The dataset is divided into training, validation, and test sets, maintaining an 8:1:1 ratio for training. On the other hand, the **PETA** dataset [3] comprises more than 8.7K pedestrians across 19K images, with diverse resolutions ranging from 17 × 39 to 169 × 365. Each pedestrian is annotated with 61 binary attributes and four multi-class attributes. However, for our current analysis, we only consider 35 attributes with a positive label ratio exceeding 5%, following the established protocol. Based on the main study [3], the dataset undergoes a random division into three splits, allocating 9.5K images for training, 1.9K images for validation, and the remaining 7.6K images for testing.

The **RAP** [14] is a collection of over 41K pedestrian images. Adhering to the original protocol by Li et al. [14], we selectively consider 51 attributes for evaluation purposes. For model evaluation, five random splits are employed, with over 33K images utilized for training and over 8K images for testing in each split. The final evaluation entails averaging the performance across all splits.

4.2 Implementation Details

We employ ViT-Base [4] encoder block to extract features which is pre-trained on ImageNet-21k, at a resolution of 224 × 224. The patch size is set to 8 × 8, 8 × 8, and 16 × 16 for granularity levels L_1, L_2 and L_3, respectively. We also set 16, 8, and 8 consecutive ViT blocks for L_1, L_2 and L_3 for level-wise feature extraction. The MLP-head for each level is a 3-layer fully-connected network with 512 and 36 neurons followed by output neurons equal to number of attributes in each level. The input image is resized to 224 × 224 and each ViT encoder block produces features of dimension $[\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i]$, where (H, W, C) are height, width, and channel of the input image. We have followed layer-wise training, with 128 batch size and 300 epochs. All heads are optimized by Adam optimizer, with $\lambda = 1$, and granularity factor are set to $\gamma_m = [0.3, 0.5, 0.8] \in m = 0, 1, 2$.

Architectural Choice: Our approach with varying patch size and ViT blocks is driven by several key considerations: (i) The number of attributes varies significantly between datasets. For example, PA100k has 26 attributes, while PETA has 61 attributes. Hence, constructing a unified architecture to accommodate such variability is not straightforward. (ii) The ViT-based architecture consists of 12 Transformer blocks [4]. Constructing a separate base architecture with 12 blocks for each attribute in PA100k may result in an impractical architecture comprising 12 times 26 blocks, leading to excessive complexity and computational demands. (iii) As noted in the ICLR 2021 paper on ViT [4], "Transformer's sequence length is inversely proportional to the square of the patch size, thus models with smaller patch sizes are computationally more expensive". Therefore, the length of the architecture needs to vary if the patch sizes vary. For L_3

attributes, where the patch size is 16 × 16, we have utilized only 8 Transformer blocks for feature extraction to maintain computational efficiency. By using different ViT configurations for different attribute levels, we ensure that the model is computationally efficient and capable of handling the varying granularity of attributes.

4.3 SOTA Comparisons

We have compared the proposed baseline with recent SOTA approaches [2,5,10,12,13,15,17–19,21–23,27]. Table 1 summarises the performance comparisons on RAP [14] and PETA [3] datasets. It can be observed that the proposed baseline achieves competitive performance through leveraging granularity-based analysis on both datasets. The experiments also reveal that ViT-based PARFormer [5] reports best recall values due to integration of attribute and viewpoint information. However, viewpoint may not always be a decisive feature, and completely relying on it may generate more false positives. This is evident when other metrics are used for comparisons.

Table 1. Prior Arts Analysis: Performance comparisons on PETA [3] and RAP [14] datasets. Best two results are shown in red and blue colors, respectively.

Method	Backbone	RAP [14]					PETA [3]				
		mA	Acc.	Prec.	Rec.	F1	mA	Acc.	Prec.	Rec.	F1
CNN + SVM [3]	VGG16	72.28	31.72	35.75	71.78	47.73	76.65	45.41	51.33	75.14	61.00
DeepMAR [12]	CaffeNet	73.79	62.02	74.92	76.21	75.56	82.89	75.07	83.68	83.14	83.41
HP-Net [18]	Inception	76.12	65.39	77.33	78.79	78.05	81.77	76.13	84.92	83.24	84.07
VeSPA [19]	Inception	77.70	67.35	**79.51**	79.67	79.59	83.45	77.73	86.18	84.81	85.49
JRL [27]	AlexNet	77.81	–	78.11	78.98	78.58	85.67	–	86.03	85.34	85.42
PgDM [13]	CaffeNet	74.31	64.57	78.86	75.90	77.35	82.97	78.08	86.86	84.68	85.76
JLPLS-PAA [22]	–	81.25	67.91	78.56	81.45	79.98	84.88	79.46	87.42	86.33	86.87
RA [32]	Inception-V3	81.16	–	79.45	79.23	79.34	86.11	–	84.69	**88.51**	86.56
ALM [23]	BN-Inception	81.87	68.17	74.71	86.48	80.16	86.30	79.52	85.65	88.09	86.85
JLAC [21]	ResNet50	83.69	69.15	79.31	82.40	80.82	86.96	80.38	**87.81**	87.09	87.45
DAFL [10]	Inception	83.72	68.18	77.41	83.39	80.29	87.07	78.88	85.78	87.03	86.40
SSC [9]	ResNet50	82.77	68.37	75.05	**87.49**	80.43	86.52	78.95	86.02	87.12	86.99
PARFormer-B [5]	Swin-B	**83.84**	**69.70**	79.24	**87.81**	**81.16**	**88.65**	**82.34**	86.89	**91.55**	**88.66**
Ours	**ViT-B**	**85.03**	**69.55**	**81.46**	**86.39**	**83.85**	**88.51**	**83.82**	**88.58**	**86.97**	**87.76**

Other notable methods such as JLAC [21], DAFL [10] show promising results on both datasets due their integration of GCN and triplet loss. Similar observations are reported on performance comparisons using the PA-100K dataset. Table 2 shows the performance of several PAR approaches. The proposed baseline achieves SOTA performance on PA-100K dataset with precision, recall, and F1 values as high as 88.13%, 90.36%, and 89.23%, respectively. It can also be observed that the overall label-based accuracy is low on PA100K [18]. It is probably due to poor annotations on PA100K dataset. This has led to smaller inter-class variations.

Table 2. Prior Method Analysis: Performance comparisons on PA100K [18]. PARFormer-B + SL is a combination Swin Transformer and semantic loss [5].

Method	mA	Acc.	Prec.	Rec.	F1
DeepMAR [12]	72.70	70.39	82.24	80.42	81.32
HP-Net [18]	74.21	72.19	82.97	82.09	82.53
JLPLS-PAA [22]	81.61	78.89	86.83	87.73	87.27
ALM [23]	80.65	77.08	84.21	88.84	86.46
JLAC [21]	82.31	79.47	87.45	87.77	87.61
Baseline [11]	81.61	79.45	**87.66**	87.59	87.62
DAFL [10]	**83.54**	80.13	87.01	89.19	**88.09**
PARFormer-B + SL [5]	**83.95**	**80.26**	87.51	**91.07**	87.69
PARFormer-B + Swin-B [5]	81.89	79.07	86.87	87.17	86.73
Ours	**82.42**	**80.17**	**88.13**	**90.36**	**89.23**

4.4 Qualitative Results

We also include qualitative results showcasing sample images from the PA100K [18] and RAP [14] datasets to ensure a fair comparison between the proposed method, DeepMAR [12], and VTB [2]. DeepMAR network is deep learning-based multi-attribute recognition network and VTB is cross-module fusion where textual information is fused with visual clues. Figure 3 depicts the prediction scores of three approaches on input samples taken from the PAR100K dataset. On the other hand, Fig. 4 illustrate performance comparison on images from RAP [14]. As depicted in the Fig. 3, the proposed method demonstrates superior robustness compared to the DeepMAR and VTB frameworks as it registered relatively higher confidence scores across a diverse range of attributes from PA100K. For instance, consider the image of a woman wearing a pink t-shirt (first column, second row) and walking with a handbag in her left hand. Despite a minor occlusion on the left side, both DeepMAR and VTB detect the handbag with a confidence score of less than 0.5, indicating low certainty regarding its presence. In contrast, the proposed method detects the handbag with a score of 0.98, showcasing its superior occlusion handling capability. Conversely, in the case of a man wearing a striped t-shirt (last example) without a hat, DeepMAR incorrectly registers a higher confidence score (0.77) for the presence of a hat, whereas the other two methods indicate very low scores.

Similar observations can be made by inspecting Fig. 4. The RAP [14] dataset offers a higher degree of diversity compared to PA100K [18], thanks to its extensive attribute set. To provide a comprehensive comparison, we intentionally selected challenging attributes to visualize results from this dataset. It is evident that DeepMAR [12] despite having its simple CNN-based architecture, the model performs well on certain attributes such as hairstyle, upper body cloth type, and face viewpoint. Conversely, the VTB [2] method primarily relies on textual descriptions of attributes. It is noticeable that complex attributes like

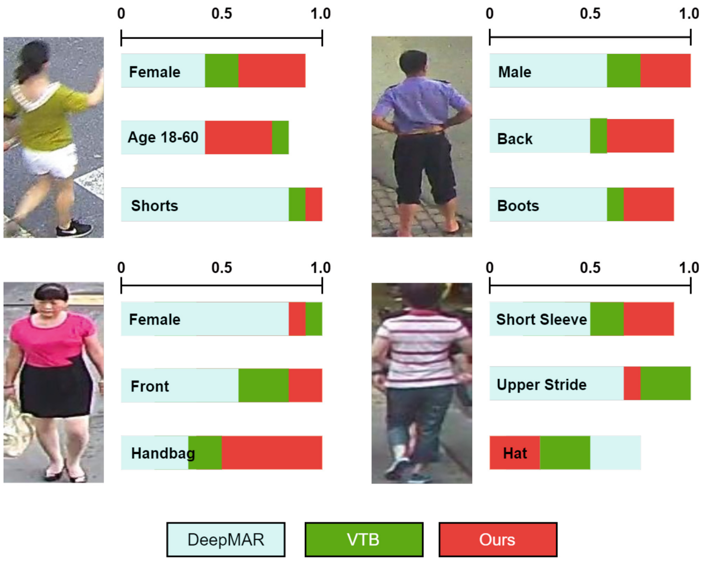


Fig. 3. Prediction Probabilities: Comparison outcomes among DeepMar [12], VTB [2], and proposed baseline on samples from1 PA100K [18] dataset samples. The bars denote the prediction probabilities between 0 to 1 and are plotted accordingly for each method.

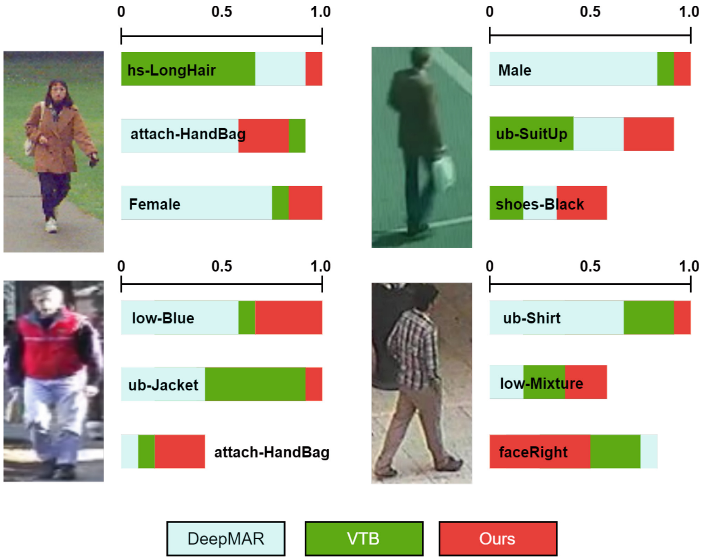


Fig. 4. Prediction Probabilities: Comparison outcomes among DeepMar [12], VTB [2], and proposed baseline on samples from RAP [14] dataset samples. The bars denote the prediction probabilities between 0 to 1 and are plotted accordingly for each method.

shoe color and *viewpoint* cannot be detected accurately due to the absence of detailed explanations.

4.5 Ablation Study

The primary components of the proposed baseline include ViT blocks for feature extraction, semantic granularity levels, and an improved loss function. To comprehend individual impact on the overall performance, we have conducted an ablation study. Initially, we have computed the performance using the basic DeepMAR [12] model with a feature extractor replaced by ResNet50. We selected the DeepMAR network as the baseline for the ablation study because of its straightforward feature extraction and classification-assisted framework. Table 3 shown performance comparisons of different components of the proposed framework on PA100K dataset. It is worth noting that DeepMAR [12] with ResNet50 as the feature extractor performs reasonably well even without the proposed spatial granularity and loss. However, it also demonstrates a notable improvement of 2.92 in mean average precision (mA) and 2.28 in precision with their inclusion. We also experimented with various feature extractors; however, we did not observe significant variation in the resulting values.

Table 3. Component-wise performance of the proposed framework on the PA100K [18].

Baseline	Granularity	Loss	mA	F1
DeepMAR [12]	×	×	72.20	81.32
DeepMAR [12]	✓	×	73.41	82.54
DeepMAR [12]	×	✓	73.69	82.77
DeepMAR [12]	✓	✓	75.12	83.60
ViT	×	×	76.48	83.50
ViT	✓	×	77.92	84.08
ViT	×	✓	78.15	85.21
ViT (Ours)	✓	✓	**82.42**	**89.23**

Since the DeepMAR [12] framework only utilizes a basic CNN structure, it may not be capable of extracting intricate features due to its limited capacity. Hence, we employed a Vision Transformer [4] to extract the latent features, leveraging its ability to capture complex patterns and relationships in the pedestrian data. By closely examining the baseline transition from DeepMAR [12] to ViT [4], it is evident that the Vision Transformer, as a feature extractor, achieved relatively higher accuracy across the dataset. However, once equipped with the proposed granularity and novel loss functions, the proposed framework achieves state-of-the-art performance on the PA100K dataset. Thus, the proposed ViT-based architecture is capable of capturing long-range dependencies alongside attribute partitioning. It thus aids in spatially focused attribute analysis that essentially contributes toward the model's ability to discern finer attribute details.

The patch sizes for granularity levels are set to 8x8, 8x8, and 16x16 for L_1, L_2, and L_3, respectively. Given that L_3 level attributes occupy minimal portions of the images, they necessitate larger patch sizes. This aligns with the statement from the Vision Transformer (ViT) paper,"Transformer's sequence length is inversely proportional to the square of the patch size", resulting in sequence lengths of 16, 8, and 8 for L_1, L_2, and L_3, respectively. We have carried out ablation experiments to assess the efficacy of these configurations. In the first variant, we have reversed the patch sizes to 16x16, 8x8, and 8x8 for L_1, L_2, and L_3, respectively. In the second variant, we have reversed the sequence lengths to 8, 8, and 16 for L_1, L_2, and L_3. The first experiment has demonstrated a significant degradation of 10.46% in mean Average Precision (mAP) on the PA100K dataset compared to the original setting. Conversely, the second setting has exhibited a slight mAP variation of 6.72%. Highest level L_1 attributes with only 8x8 patch sizes necessitate global-level feature analysis, thus requiring a higher number of transformer blocks. When the patch size is reversed to 16x16, the transformer fails to capture global features effectively due to the insufficiently small patch size. This issue is consistent across other levels as well.

5 Conclusion and Future Work

In response to the challenges presented by current Pedestrian Attribute Recognition (PAR) datasets and approches, we have devised a comprehensive approach. Our suggested framework involves employing a robust baseline model utilizing Vision Transformer (ViT) blocks. We have further enhanced this framework by categorizing attributes into three spatial granularity levels, exploiting hierarchical feature extraction to capture both global and local visual cues effectively. Additionally, we have introduced a novel loss function designed specifically to mitigate the inherent data imbalance prevalent in PAR datasets. The proposed solution approach not only ensures more stable model training but also achieve superior performance in accurately recognizing pedestrian attributes. We conducted extensive experiments on three widely used Pedestrian Attribute Recognition (PAR) datasets: RAP, PA100K, and PETA. The results demonstrate that our proposed approach has achieved significant improvements across all three datasets, depicting its effectiveness and robustness in addressing the challenges inherent in PAR. One limitation of the proposed approach is its dependence on pre-defined spatial granularity levels for attribute classification. This fixed granularity may not fully capture the diverse and nuanced visual characteristics present in real-world pedestrian images. Consequently, the model's ability to adapt to novel or unexpected attribute variations might be limited. To address this limitation, future research could explore dynamic or adaptive spatial granularity schemes that allow the model to adjust its feature extraction process based on the specific attributes and context present in each image.

Acknowledgement. This research was supported by Development of Proactive Crowd Density Management Platform Based on Spatiotemporal Multimodal Data

Analysis and High-Precision Digital Twin Simulation Program through the Korea Institute of Police Technology (KIPoT) funded by the Korean National Police Agency & Ministry of the Interior and Safety (RS-2024-00405100), and Korea Institute of Science and Technology (KIST) Institutional Program (Project No.2E33001). Morever, the implementation phase of the study is executed at IIT Bhubaneswar under the Project code CP438.

References

1. Bui, D.C., Le, T.V., Ngo, B.H.: C2t-net: channel-aware cross-fused transformer-style networks for pedestrian attribute recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 351–358 (2024)
2. Cheng, X., Jia, M., Wang, Q., Zhang, J.: A simple visual-textual baseline for pedestrian attribute recognition. IEEE Trans. Circuit Syst. Video Technol. (2022)
3. Deng, Y., Luo, P., Loy, C.C., Tang, X.: Pedestrian attribute recognition at far distance. ACM Int. Conf. Multimed. (2014)
4. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Fan, X., Zhang, Y., Lu, Y., Wang, H.: Parformer: transformer-based multi-task network for pedestrian attribute recognition. IEEE Trans. Circuit Syst. Video Technol. (2023)
6. Huang, Y., Wu, Q., Xu, J., Zhong, Y., Zhang, Z.: Clothing status awareness for long-term person re-identification. In: International Conference on Computer Vision, pp. 11895–11904 (2021)
7. Huang, Y., Wu, Q., Xu, J., Zhong, Y., Zhang, Z.: Unsupervised domain adaptation with background shift mitigating for person re-identification. Int. J. Comput. Vis. **129**, 2244 (2021)
8. Jia, J., Chen, X., Huang, K.: Spatial and semantic consistency regularizations for pedestrian attribute recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 962–971 (2021)
9. Jia, J., Chen, X., Huang, K.: Spatial and semantic consistency regularizations for pedestrian attribute recognition. In: International Conference on Computer Vision (2021)
10. Jia, J., Gao, N., He, F., Chen, X., Huang, K.: Learning disentangled attribute representations for robust pedestrian attribute recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
11. Jia, J., Huang, H., Chen, X., Huang, K.: Rethinking of Pedestrian Attribute Recognition: a Reliable Evaluation Under Zero-Shot Pedestrian Identity Setting (2021). arXiv preprint arXiv:2107.03576
12. Li, D., Chen, X., Huang, K.: Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In: IAPR Asian Conference on Pattern Recognition (2015)
13. Li, D., Chen, X., Zhang, Z., Huang, K.: Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. In: IEEE International Conference on Multimedia and Expo (ICME) (2018)
14. Li, D., Zhang, Z., Chen, X., Huang, K.: A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. IEEE Trans. Image Process. **28**, 1575 (2018)

15. Li, Q., Zhao, X., He, R., Huang, K.: Visual-semantic graph reasoning for pedestrian attribute recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
16. Lin, Y., et al.: Improving person re-identification by attribute and identity learning. Pattern Recognition (2019)
17. Liu, P., Liu, X., Yan, J., Shao, J.: Localization guided learning for pedestrian attribute recognition. Brit. Mach. Vis. Conf. (2018)
18. Liu, X., et al.: Hydraplus-net: Attentive deep features for pedestrian analysis. Int. Conf. Comput. Vis. (2017)
19. Sarfraz, M.S., Schumann, A., Wang, Y., Stiefelhagen, R.: Deep view-sensitive pedestrian attribute inference in an end-to-end model (2017). arXiv preprint arXiv:1707.06089
20. Specker, A., Cormier, M., Beyerer, J.: UPAR: Unified pedestrian attribute recognition and person retrieval. In: Winter Conference on Applications of Computer Vision (2023)
21. Tan, Z., Yang, Y., Wan, J., Guo, G., Li, S.Z.: Relation-aware pedestrian attribute recognition with graph convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
22. Tan, Z., Yang, Y., Wan, J., Hang, H., Guo, G., Li, S.Z.: Attention-based pedestrian attribute analysis. IEEE Trans. Image Process (2019)
23. Tang, C., Sheng, L., Zhang, Z., Hu, X.: Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
24. Tang, S., Chen, C., et al.: Humanbench: towards general human-centric perception with projector assisted pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
25. Thakare, K.V., Raghuwanshi, Y., Dogra, D.P., Choi, H., Kim, I.J.: Dyannet: a scene dynamicity guided self-trained video anomaly detection network. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 5541–5550 (2023)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł, Polosukhin, I.: Attention is all you need. Adv. Neural Inform. Process. Syst. **30**, 1 (2017)
27. Wang, J., Zhu, X., Gong, S., Li, W.: Attribute recognition by joint recurrent learning of context and correlation. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
28. Wang, S., Duan, Y., Ding, H., Tan, Y.P., Yap, K.H., Yuan, J.: Learning transferable human-object interaction detector with natural language supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(2022)
29. Wu, G., Zhu, X., Gong, S.: Learning hybrid ranking representation for person re-identification. Pattern Recog. **121**, 108239 (2022)
30. Yan, Y., Xu, Y., Xue, J.H., Lu, Y., Wang, H., Zhu, W.: Drop loss for person attribute recognition with imbalanced noisy-labeled samples. IEEE Trans. Cybernet. **53**, 7071 (2023)
31. Yang, Y., Tan, Z., Tiwari, P., Pandey, H.M., Wan, J., Lei, Z., Guo, G., Li, S.Z.: Cascaded split-and-aggregate learning with feature recombination for pedestrian attribute recognition. Int. J. Comput. Vis. **129**, 2731–2744 (2021)
32. Zhao, X., Sang, L., Ding, G., Han, J., Di, N., Yan, C.: Recurrent attention model for pedestrian attribute recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)

Born-Again Multi-task Self-training for Multi-task Facial Emotion Recognition

Ryo Masumura(✉), Akihiko Takashima, Satoshi Suzuki, and Shota Orihashi

NTT Corporation, Yokosuka, Japan
ryo.masumura@ntt.com

Abstract. In this paper, we propose novel learning methods for multi-task facial emotion recognition that simultaneously performs multiple facial emotion recognition tasks. It is often difficult to collect datasets with multiple completely annotated targets (i.e., fundamental emotion, valence-arousal, action unit intensity, etc.), so a technology is required that constructs a multi-task facial emotion recognition model from multiple single-target annotated datasets. To this end, previous studies have introduced a method called multi-task self-training, in which the multi-task learning is performed by supplementing missing targets with pseudo-targets generated using single-task models. However, the pseudo-targets are often not precise enough, so effective multi-task learning cannot be performed. To address this problem, our proposed method, called born-again multi-task self-training, refines the pseudo-targets via iterative born-again steps of the multi-task model, i.e., the pseudo-targets are regenerated using the pre-trained multi-task model. In addition, to enhance the born-again steps, the proposed method temporarily creates single-task models via fine-tuning of the pre-trained multi-task model. The temporal single-task models effectively regenerate precise pseudo-targets. In our experiments with three facial emotion recognition tasks, we demonstrate that the proposed method outperforms the conventional multi-task self-training.

Keywords: multi-task model · facial emotion recognition · multi-task self-training · born-again · psudo-targets

1 Introduction

Analysis of facial information is crucial as human face images hold a wealth of information, particularly facial expressions, that is linked to human emotions [1]. There are several prominent tasks in the field of facial emotion recognition, including facial expression classification [2], estimation of action units that represent individual facial muscle movements [3], and estimation of valence-arousal, which is a two-dimensional value based on the circumplex model of emotion [4]. In general, these tasks are individually modeled as task-specific models, i.e., single-task models. In recent years, approaches have been developed to perform multiple facial emotion recognition tasks with one multi-task model [5],

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 94–108, 2025.
https://doi.org/10.1007/978-3-031-78444-6_7

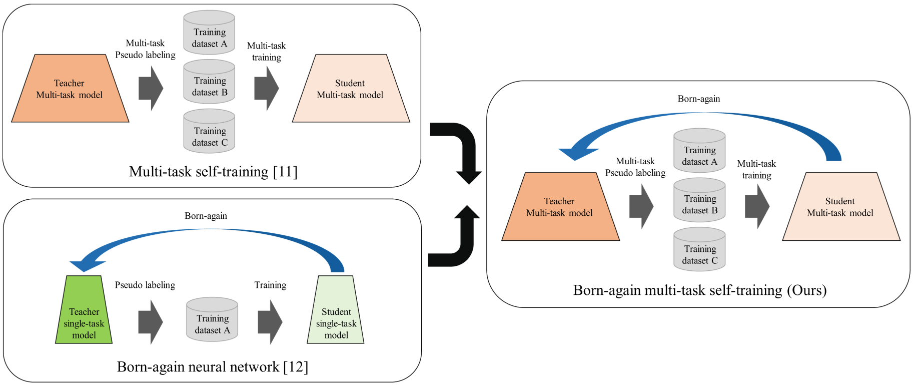


Fig. 1. Key idea of born-again multi-task self-training

which has the advantage of enhancing the knowledge of each task during learning and reducing computation costs during inference. Therefore, this paper aims to develop methods for constructing a multi-task model that can effectively perform multiple facial emotion recognition tasks using one multi-task model.

In previous studies, multi-task learning methods to build multi-task models have been investigated [6–9]. Basically, multi-task learning requires a dataset with multiple completely annotated targets. Unfortunately, completely annotating a dataset is costly and not practical under realistic conditions. Therefore, a multi-task model needs to be built using easily obtainable single-target annotated datasets. The most basic approach under this condition is to calculate the loss only for the output part of the corresponding task when data for a certain task is inputted [10]. This approach cannot learn the relationships between tasks from a single piece of data. Recently, multi-task self-training [11] has been used to perform multi-task learning from multiple single-target annotated datasets. This method first trains single-task models as teachers for each task and uses them to supplement missing targets with pseudo-targets. Then, a multi-task model is trained from pseudo-completely annotated datasets via the pseudo-target annotation.

However, conventional multi-task self-training methods often fail to produce the synergistic effects of multi-task learning because the pseudo-target annotated by the single-task teacher models are not necessarily accurate or informative. If the accuracy of pseudo-targets is poor, it is expected to negatively impact accuracy instead of transferring knowledge between tasks. Therefore, multi-task learning with single-target annotated datasets requires improved accuracy of pseudo-targets.

In this paper, we propose a born-again multi-task self-training method for effective multi-task learning from multiple single-target annotated datasets. The proposed method extends the conventional multi-task self-training to improve the qualities of pseudo-targets. Figure 1 shows the key idea of born-again multi-task self-training. The idea is to use the constructed multi-task model as a teacher to iteratively refine the multi-task model. This can refine the information of the pseudo-targets and perform more sophisticated multi-task self-training. Born-again steps have been studied in born-again neural networks [12] in which knowledge distillation using the same model structure is repeated. Different from the previous study, this paper is the first to perform multi-task self-training while being born-again. Furthermore, to enhance the effect of the born-again steps, the proposed method temporarily creates single-task models specialized for each task via fine-tuning of the pre-trained multi-task model. The temporal single-task models effectively regenerate precise pseudo-targets compared with the pre-trained multi-task model. This is because the fine-tuned temporal single-task models can utilize the good parts of multi-task knowledge through multi-task learning and task-specific knowledge at the same time. We expect that improved accuracy of pseudo-targets will promote improved multi-task model training.

In our experiments, we deal with a facial expression classification task, action unit intensity estimation task, and valence-arousal estimation task as multi-tasks. Our experiments show that the proposed method with iterative born-again steps outperforms conventional multi-task learning methods.

2 Definition

In this paper, we handle three facial emotion recognition tasks: facial expression classification (EXPR), action unit intensity estimation (AU), and valence-arousal estimation (VA). This section defines the single-target annotated dataset, modeling, and loss functions for the supervised learning.

2.1 Single-Target Annotated Datasets

The single-target annotated dataset for each task is defined as

$$\mathcal{D}^{(i)} = \{(\boldsymbol{x}_n^{(i)}, \boldsymbol{y}_n^{(i)}) \mid n \in \{1, \cdots, N^{(i)}\}\}, \tag{1}$$

where, $\boldsymbol{x}_n^{(i)}$ is the input image and $\boldsymbol{y}_n^{(i)} = [y_{n,1}^{(i)}, \cdots, y_{n,K^{(i)}}^{(i)}]^\top$ is the ground-truth target for the n-th sample in the i-th single-target annotated dataset. $N^{(i)}$ represents the number of samples for the i-th dataset and $K^{(i)}$ represents the number of elements in the i-th task's target.

Each input image is annotated with only one task. $i \in \{1, 2, 3\}$ are the indices for each task, where $i = 1$, $i = 2$, and $i = 3$ refer to EXPR, AU, VA tasks, respectively. In EXPR task, $\boldsymbol{y}_n^{(1)}$ is a one-hot vector. In an AU task, $\boldsymbol{y}_n^{(2)}$ has multiple action unit elements, each of which has a value in the range $[0, 5]$. In VA task, $\boldsymbol{y}_n^{(3)}$ has two elements, the valence value and the arousal value, each of which has a value in the range $[-1, 1]$.

2.2 Modeling

This paper uses single-task models that individually have a task-specific input layer and a task-specific output layer, and a multi-task model that has a task-agnostic input layer and multiple task-specific output layers. We denote the output of a single-task model for the i-th task by

$$\hat{\boldsymbol{y}}^{(i)} = \mathcal{S}^{(i)}(\boldsymbol{x}; \theta^{(i)}), \tag{2}$$

where $\mathcal{S}^{(i)}(\cdot)$ is the model function and $\theta^{(i)}$ is its model parameter of the i-th single-task model. We denote the output for the i-th task of a multi-task model by

$$\hat{\boldsymbol{y}}^{(i)} = \mathcal{M}^{(i)}(\boldsymbol{x}; \theta), \tag{3}$$

where $\mathcal{M}^{(i)}(\cdot)$ is the model function that predicts the i-th output and θ is a shared model parameter.

2.3 Loss Functions

For each task, we define loss functions, which are cross-entropy loss for the EXPR task and mean squared error for AU and VA tasks. When targets are available for each task, we define loss functions for supervised learning as

$$\mathcal{L}^{(1)}(\boldsymbol{y}^{(1)}, \hat{\boldsymbol{y}}^{(1)}) = -\sum_{k=1}^{K^{(1)}} y_k^{(1)} \log \hat{y}_k^{(1)}, \tag{4}$$

$$\mathcal{L}^{(2)}(\boldsymbol{y}^{(2)}, \hat{\boldsymbol{y}}^{(2)}) = \frac{1}{K^{(2)}} \sum_{k=1}^{K^{(2)}} (y_k^{(2)} - \hat{y}_k^{(2)})^2, \tag{5}$$

$$\mathcal{L}^{(3)}(\boldsymbol{y}^{(3)}, \hat{\boldsymbol{y}}^{(3)}) = \frac{1}{K^{(3)}} \sum_{k=1}^{K^{(3)}} (y_k^{(3)} - \hat{y}_k^{(3)})^2, \tag{6}$$

where $\hat{\boldsymbol{y}}_n^{(i)} = [y_1^{(i)}, \cdots, y_{K^{(i)}}^{(i)}]^\top$ is the outputs of the i-th task's model. The outputs can be computed from either the single- or multi-task model.

3 Baseline Methods

This section details baseline training methods that train single-task models.

3.1 Task-Specific Training

Task-specific training, i.e., supervised learning, utilizes single-target annotated data for building a single-task model. Figure 2 shows an overview of task-specific training. To train a single-task model for the i-th task, the loss function is defined as

$$\mathcal{L}(\theta^{(i)}) = \sum_{n=1}^{N^{(i)}} \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{S}^{(i)}(\boldsymbol{x}_n^{(i)}; \theta^{(i)})). \tag{7}$$

The model parameter can be optimized by minimizing the loss function.

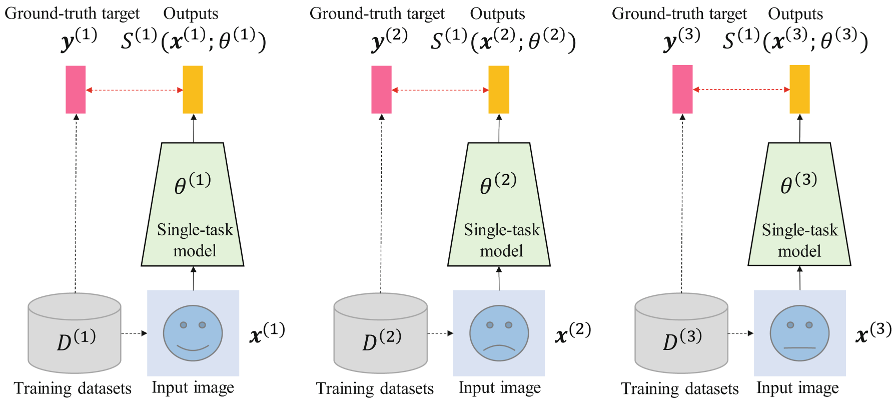


Fig. 2. An overview of task-specific training

3.2 Task-Specific Self-training

Task-specific self-training utilizes not only single-target annotated data but also pseudo-target annotated data [13,14]. The pseudo-target is obtained using a pre-trained single-task model as a teacher. Thus, pseudo-target is annotated against an input image in another-task's datasets. The pre-trained single-task model is attained by task-specific training detailed in Sect. 3.1. Figure 3 shows an overview of task-specific self-training. In this case, $\theta^{(1)}$ is trained with task-specific self-training. To train a single-task model for the i-th task, the loss function is defined as

$$\mathcal{L}(\theta^{(i)}) = \sum_{n=1}^{N^{(i)}} \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{S}^{(i)}(\boldsymbol{x}_n^{(i)}; \theta^{(i)})) \\ + \sum_{m \neq i} \sum_{n=1}^{N^{(m)}} \mathcal{L}^{(i)}(\mathcal{S}^{(i)}(\boldsymbol{x}_n^{(m)}; \hat{\theta}^{(i)}), \mathcal{S}^{(i)}(\boldsymbol{x}_n^{(m)}; \theta^{(i)})), \tag{8}$$

where $\hat{\theta}^{(i)}$ is the frozen pre-trained parameters. Note that $\hat{\theta}^{(i)}$ is trained on Eq. (7).

4 Conventional Method

This section describes conventional multi-task self-training [11]. This training has two steps. First, single-task models are trained as teachers from a single-target annotated dataset for each task. Next, a multi-task model is trained from pseudo-completely annotated datasets that are created using the single-task teacher models. Figure 4 shows an overview of multi-task self-training.

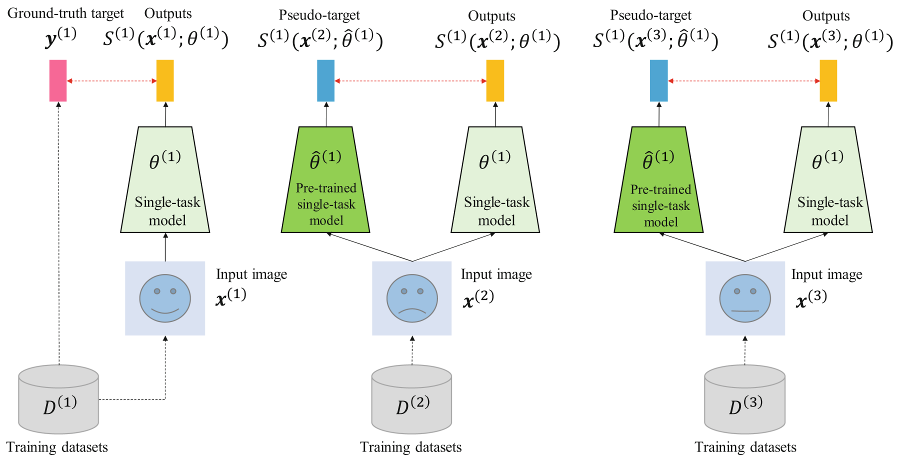

Ground-truth target
$y^{(1)}$
Outputs
$S^{(1)}(x^{(1)};\theta^{(1)})$
Pseudo-target
$S^{(1)}(x^{(2)};\hat{\theta}^{(1)})$
Outputs
$S^{(1)}(x^{(2)};\theta^{(1)})$
Pseudo-target
$S^{(1)}(x^{(3)};\hat{\theta}^{(1)})$
Outputs
$S^{(1)}(x^{(3)};\theta^{(1)})$
$\theta^{(1)}$ Single-task model
$\hat{\theta}^{(1)}$ Pre-trained single-task model
$\theta^{(1)}$ Single-task model
$\hat{\theta}^{(1)}$ Pre-trained single-task model
$\theta^{(1)}$ Single-task model
Input image $x^{(1)}$
Input image $x^{(2)}$
Input image $x^{(3)}$
$D^{(1)}$
Training datasets
$D^{(2)}$
Training datasets
$D^{(3)}$
Training datasets

Fig. 3. An overview of task-specific self-training

In the first step, a single-task model is built from a single-target annotated dataset for each task. This is achieved by task-specific training detailed in Sect. 3.1. In the second step, a multi-task model is trained using the teacher models. We define the loss function to train the multi-task model as

$$\mathcal{L}(\theta) = \sum_{i=1}^{3} \sum_{n=1}^{N^{(i)}} \Big\{ \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{M}^{(i)}(\boldsymbol{x}_n^{(i)};\theta)) \\ + \sum_{m \neq i} \mathcal{L}^{(m)}(\mathcal{S}^{(m)}(\boldsymbol{x}_n^{(i)};\hat{\theta}^{(m)}), \mathcal{M}^{(m)}(\boldsymbol{x}_n^{(i)};\theta)) \Big\}, \tag{9}$$

where the model parameters $\hat{\theta}^{(1)}$, $\hat{\theta}^{(2)}$, $\hat{\theta}^{(3)}$ are the fixed parameters trained in Eq. (7). Note that the loss weights to take the importance of each loss term into consideration are omitted in Eq. (9) for simplicity.

5 Proposed Methods

This section details proposed born-again multi-task self-training. In the proposed method, the multi-task model is first constructed using the conventional multi-task self-learning framework formulated in Eq. (9). Then, the multi-task model is leveraged as a teacher for generating pseudo-targets. We present two methods: simple born-again multi-task self-training and task-specific born-again multi-task self-training.

5.1 Simple Born-Again Multi-task Self-training

In simple born-again multi-task self-training, we use the pre-trained multi-task model $\hat{\theta}$ as a teacher to build a next-generation multi-task model θ. Figure 5

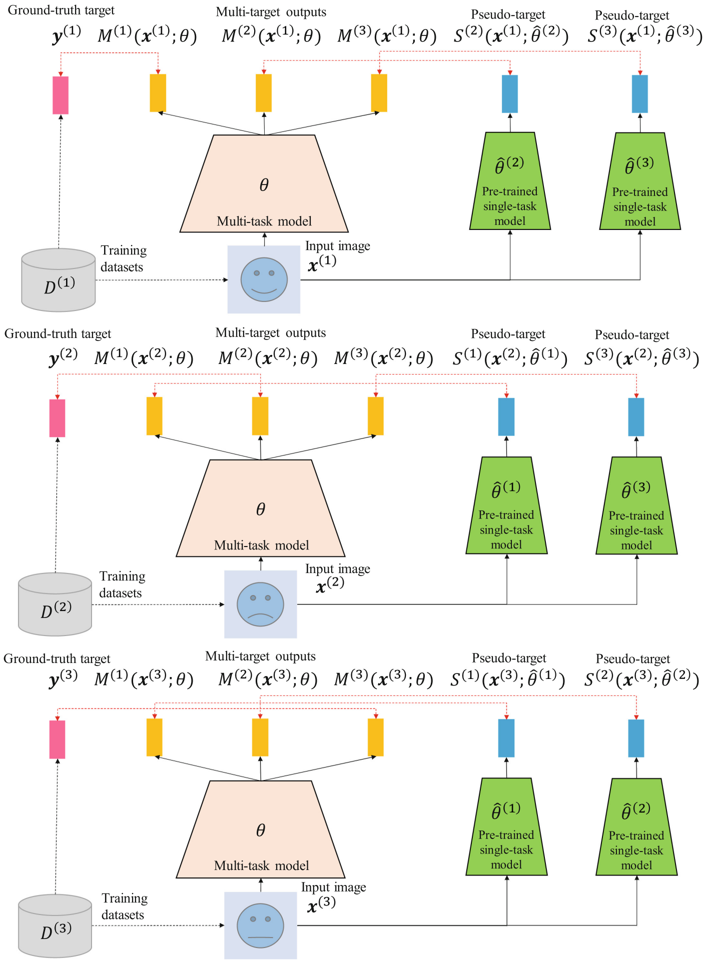

Ground-truth target
$\boldsymbol{y}^{(1)}$
Multi-target outputs
$M^{(1)}(\boldsymbol{x}^{(1)};\theta)$
$M^{(2)}(\boldsymbol{x}^{(1)};\theta)$
$M^{(3)}(\boldsymbol{x}^{(1)};\theta)$
Pseudo-target
$S^{(2)}(\boldsymbol{x}^{(1)};\hat{\theta}^{(2)})$
Pseudo-target
$S^{(3)}(\boldsymbol{x}^{(1)};\hat{\theta}^{(3)})$
θ
Multi-task model
$\hat{\theta}^{(2)}$
Pre-trained single-task model
$\hat{\theta}^{(3)}$
Pre-trained single-task model
Training datasets
$D^{(1)}$
Input image
$\boldsymbol{x}^{(1)}$
Ground-truth target
$\boldsymbol{y}^{(2)}$
Multi-target outputs
$M^{(1)}(\boldsymbol{x}^{(2)};\theta)$
$M^{(2)}(\boldsymbol{x}^{(2)};\theta)$
$M^{(3)}(\boldsymbol{x}^{(2)};\theta)$
Pseudo-target
$S^{(1)}(\boldsymbol{x}^{(2)};\hat{\theta}^{(1)})$
Pseudo-target
$S^{(3)}(\boldsymbol{x}^{(2)};\hat{\theta}^{(3)})$
θ
Multi-task model
$\hat{\theta}^{(1)}$
Pre-trained single-task model
$\hat{\theta}^{(3)}$
Pre-trained single-task model
Training datasets
$D^{(2)}$
Input image
$\boldsymbol{x}^{(2)}$
Ground-truth target
$\boldsymbol{y}^{(3)}$
Multi-target outputs
$M^{(1)}(\boldsymbol{x}^{(3)};\theta)$
$M^{(2)}(\boldsymbol{x}^{(3)};\theta)$
$M^{(3)}(\boldsymbol{x}^{(3)};\theta)$
Pseudo-target
$S^{(1)}(\boldsymbol{x}^{(3)};\hat{\theta}^{(1)})$
Pseudo-target
$S^{(2)}(\boldsymbol{x}^{(3)};\hat{\theta}^{(2)})$
θ
Multi-task model
$\hat{\theta}^{(1)}$
Pre-trained single-task model
$\hat{\theta}^{(2)}$
Pre-trained single-task model
Training datasets
$D^{(3)}$
Input image
$\boldsymbol{x}^{(3)}$

Fig. 4. An overview of multi-task self-training

shows an overview of multi-task self-training. We define the loss function to train the next-generation multi-task model as

$$\mathcal{L}(\theta) = \sum_{i=1}^{3}\sum_{n=1}^{N^{(i)}} \Big\{ \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{M}^{(i)}(\boldsymbol{x}_n^{(i)};\theta)) + \sum_{m\neq i} \mathcal{L}^{(m)}(\mathcal{M}^{(m)}(\boldsymbol{x}_n^{(i)};\hat{\theta}), \mathcal{M}^{(m)}(\boldsymbol{x}_n^{(i)};\theta)) \Big\}, \tag{10}$$

where $\hat{\theta}$ is fixed non-trainable parameters.

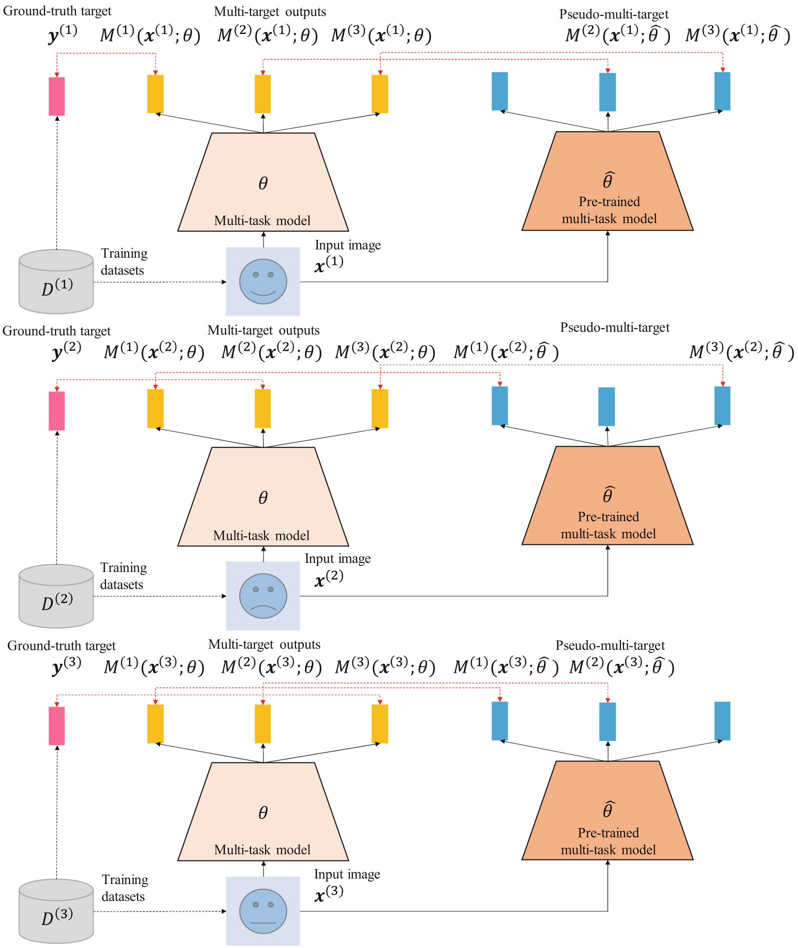

Ground-truth target
Multi-target outputs
Pseudo-multi-target
$y^{(1)}$
$M^{(1)}(x^{(1)};\theta)$
$M^{(2)}(x^{(1)};\theta)$
$M^{(3)}(x^{(1)};\theta)$
$M^{(2)}(x^{(1)};\hat{\theta})$
$M^{(3)}(x^{(1)};\hat{\theta})$
θ
Multi-task model
$\hat{\theta}$
Pre-trained multi-task model
Training datasets
$D^{(1)}$
Input image
$x^{(1)}$
Ground-truth target
Multi-target outputs
Pseudo-multi-target
$y^{(2)}$
$M^{(1)}(x^{(2)};\theta)$
$M^{(2)}(x^{(2)};\theta)$
$M^{(3)}(x^{(2)};\theta)$
$M^{(1)}(x^{(2)};\hat{\theta})$
$M^{(3)}(x^{(2)};\hat{\theta})$
θ
Multi-task model
$\hat{\theta}$
Pre-trained multi-task model
Training datasets
$D^{(2)}$
Input image
$x^{(2)}$
Ground-truth target
Multi-target outputs
Pseudo-multi-target
$y^{(3)}$
$M^{(1)}(x^{(3)};\theta)$
$M^{(2)}(x^{(3)};\theta)$
$M^{(3)}(x^{(3)};\theta)$
$M^{(1)}(x^{(3)};\hat{\theta})$
$M^{(2)}(x^{(3)};\hat{\theta})$
θ
Multi-task model
$\hat{\theta}$
Pre-trained multi-task model
Training datasets
$D^{(3)}$
Input image
$x^{(3)}$

Fig. 5. An overview of simple born-again multi-task self-training

In the first born-again step, θ trained on conventional multi-task self-training is used as $\hat{\theta}$. The born-again can be iterated by replacing the trained next-generation multi-task model as $\hat{\theta}$ in Eq. (10).

5.2 Task-Specific Born-Again Multi-task Self-training

In task-specific born-again multi-task self-training, we temporarily build single-task models $\theta_{\text{tmp}}^{(1)}$, $\theta_{\text{tmp}}^{(2)}$, $\theta_{\text{tmp}}^{(3)}$ using the pre-trained multi-for generating pseudo-targets. Figure 6 shows an overview of building the temporal single-task models. In this case, $\theta_{\text{tmp}}^{(1)}$ is trained. To build the i-th temporal single-task model, we define the loss function as

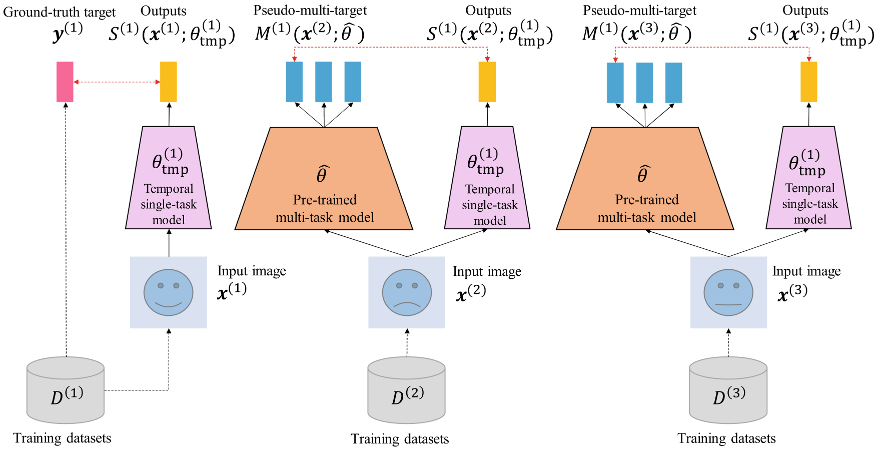

Ground-truth target
$y^{(1)}$
Outputs
$S^{(1)}(x^{(1)}; \theta_{\text{tmp}}^{(1)})$
Pseudo-multi-target
$M^{(1)}(x^{(2)}; \hat{\theta})$
Outputs
$S^{(1)}(x^{(2)}; \theta_{\text{tmp}}^{(1)})$
Pseudo-multi-target
$M^{(1)}(x^{(3)}; \hat{\theta})$
Outputs
$S^{(1)}(x^{(3)}; \theta_{\text{tmp}}^{(1)})$
$\theta_{\text{tmp}}^{(1)}$
Temporal single-task model
$\hat{\theta}$
Pre-trained multi-task model
Input image
$x^{(1)}$
$x^{(2)}$
$x^{(3)}$
$D^{(1)}$
$D^{(2)}$
$D^{(3)}$
Training datasets

Fig. 6. An overview of building a temporal single-task model in task-specific born-again multi-task self-training

$$\begin{aligned}\mathcal{L}(\theta_{\text{tmp}}^{(i)}) &= \sum_{n=1}^{N^{(i)}} \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{S}^{(i)}(\boldsymbol{x}_n^{(i)}; \theta_{\text{tmp}}^{(i)})) \\ &+ \sum_{m \neq i} \sum_{n=1}^{N^{(m)}} \mathcal{L}^{(i)}(\mathcal{M}^{(m)}(\boldsymbol{x}_n^{(i)}; \hat{\theta}), \mathcal{S}^{(i)}(\boldsymbol{x}_n^{(i)}; \theta_{\text{tmp}}^{(i)})),\end{aligned} \tag{11}$$

where $\hat{\theta}$ is fixed non-trainable parameters. In the first born-again step, model parameters trained on conventional multi-task self-training is used as $\hat{\theta}$. The temporal single-task models effectively regenerate precise pseudo-targets compared with the pre-trained multi-task model. Therefore, the pre-trained temporal single-task models $\hat{\theta}_{\text{tmp}}^{(1)}$, $\hat{\theta}_{\text{tmp}}^{(2)}$, $\hat{\theta}_{\text{tmp}}^{(3)}$ are used as teachers for building the next-generation multi-task model θ. Figure 7 shows an overview of task-specific born-again multi-task self-training. To train the next-generation multi-task model, the loss function is defined as

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{i=1}^{3} \sum_{n=1}^{N^{(i)}} \Big\{ \mathcal{L}^{(i)}(\boldsymbol{y}_n^{(i)}, \mathcal{M}^{(i)}(\boldsymbol{x}_n^{(i)}; \theta)) \\ &+ \sum_{m \neq i} \mathcal{L}^{(m)}(\mathcal{S}^{(m)}(\boldsymbol{x}_n^{(i)}; \hat{\theta}_{\text{tmp}}^{(i)}), \mathcal{M}^{(m)}(\boldsymbol{x}_n^{(i)}; \theta)) \Big\},\end{aligned} \tag{12}$$

where $\hat{\theta}_{\text{tmp}}^{(1)}$, $\hat{\theta}_{\text{tmp}}^{(2)}$, $\hat{\theta}_{\text{tmp}}^{(3)}$ are fixed non-trainable parameters. The born-again can be iterated by replacing the trained next-generation multi-task model as $\hat{\theta}$ in Eq. (11).

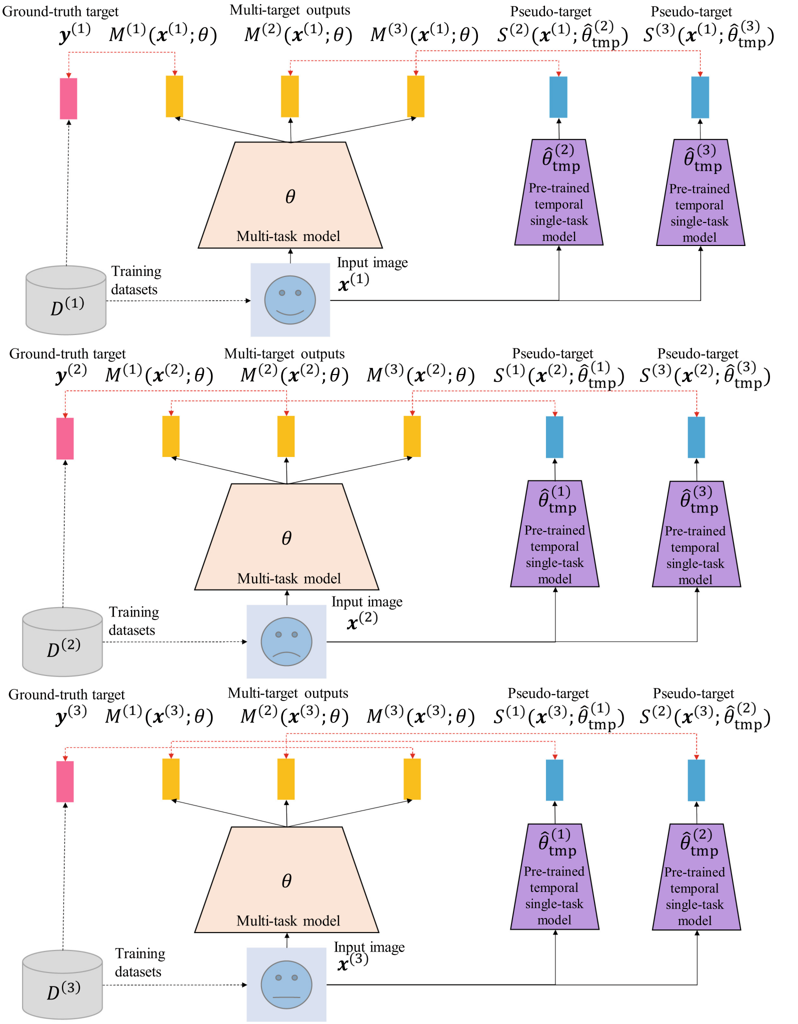

Ground-truth target
Multi-target outputs
Pseudo-target
Pseudo-target
$y^{(1)}$ $M^{(1)}(x^{(1)};\theta)$ $M^{(2)}(x^{(1)};\theta)$ $M^{(3)}(x^{(1)};\theta)$ $S^{(2)}(x^{(1)};\hat{\theta}^{(2)}_{tmp})$ $S^{(3)}(x^{(1)};\hat{\theta}^{(3)}_{tmp})$
θ
Multi-task model
$\hat{\theta}^{(2)}_{tmp}$ Pre-trained temporal single-task model
$\hat{\theta}^{(3)}_{tmp}$ Pre-trained temporal single-task model
$D^{(1)}$
Training datasets
Input image $x^{(1)}$
$y^{(2)}$ $M^{(1)}(x^{(2)};\theta)$ $M^{(2)}(x^{(2)};\theta)$ $M^{(3)}(x^{(2)};\theta)$ $S^{(1)}(x^{(2)};\hat{\theta}^{(1)}_{tmp})$ $S^{(3)}(x^{(2)};\hat{\theta}^{(3)}_{tmp})$
$\hat{\theta}^{(1)}_{tmp}$ Pre-trained temporal single-task model
$D^{(2)}$
Input image $x^{(2)}$
$y^{(3)}$ $M^{(1)}(x^{(3)};\theta)$ $M^{(2)}(x^{(3)};\theta)$ $M^{(3)}(x^{(3)};\theta)$ $S^{(1)}(x^{(3)};\hat{\theta}^{(1)}_{tmp})$ $S^{(2)}(x^{(3)};\hat{\theta}^{(2)}_{tmp})$
$D^{(3)}$
Input image $x^{(3)}$

Fig. 7. An overview of task-specific born-again multi-task self-training

6 Experiments

To verify the effectiveness of the proposed method, we evaluated its performance in facial expression classification, action unit intensity estimation, and valence-arousal estimation tasks.

6.1 Datasets

We used multiple single-target annotated datasets for evaluation.

- **EXPR datasets:** We used two datasets, FER2013 [15] and RAF-DB [16], both annotated with seven categories: neutral, happy, sad, angry, fearful, disgusted, and surprised. Both datasets used published training and test sets of about 41,000 and 6,600 images, respectively.
- **AU datasets:** We used the DISFA dataset [17], which consists of video data from 27 subjects annotated with 12 elements of AU intensity at the frame level. We used 21 subjects as our training set and the remaining 6 subjects as our test set, which consisted of about 96,000 and 29,000 images, respectively.
- **VA datasets:** We used the AffectNet dataset [18], which is annotated with valence-arousal values. As a reminder, AffectNet is also annotated with targets for the facial expression classification task, but we did not use them. To ensure consistency with other tasks, we randomly sampled 50,000 images from the published training set, and we used 4,500 images for testing, published as a test set.

Each dataset has different face image cropping due to variations in their respective domains. For effective multi-task learning, a common face alignment is necessary. Therefore, we used a landmark detector [19] to align the faces with the same criteria and compensate for facial tilt. The face images were cropped and resized to 256×256.

6.2 Setups

We evaluated single-task models with baseline methods (task-specific training and task-specific self-training [13,14]), a multi-task model with conventional multi-task self-training [11], and multi-task models with proposed methods (simple born-again multi-task self-training and task-specific born-again multi-task self-training).

For the single-task models and multi-task models, the MobileNetV3 architecture [20] was used for the backbone network, which is a 13-layer convolutional neural network (CNN). After the global average pooling layer, two fully connected layers with 256 dimensions and am output layer are added. For the EXPR task, a softmax layer was used as the output layer. For the AU and VA tasks, a linear layer was used as the output layer. In the multi-task model, the MobileNetV3 architecture serves as a shared backbone network. The model parameters in the MobilenetV3 were pre-trained with the VGGFace2 dataset [21]. To construct the single-task and multi-task models, the mini-batch size was set to 128, and we used Adam [22] for optimization. The training steps were stopped on the basis of early stopping using a part of the training sets. Loss weights in multi-task self-training were determined by grid search.

6.3 Results

The experimental results are shown in Table 1. The EXPR task used accuracy, the AU task used the Intra-Class Correlation ICC (3,1) [23], and the VA task used the concordance correlation coefficient [24]. As an evaluation score for overall tasks, we defined the summed score of each task. "*Overall-S*" represents an overall score when using three single-task models, and "*Overall-M*" represents that when using a multi-task model. Note that "iter." represents the number of performing multi-task self-training in the conventional and proposed methods. Thus, the number of performing multi-task self-training in the conventional method is 1.

Table 1. Experimental results of baseline, conventional and proposed methods

Single-task model with baseline method		EXPR	AU	*VA*	*Overall-S*
Task-specific training		0.715	0.413	0.442	1.570
Task-specific self-training [13,14]		0.728	0.435	0.453	1.616
Multi-task model with conventional method	iter.	EXPR	AU	*VA*	*Overall-M*
Multi-task self-training [11]	1	0.719	0.430	0.462	1.611
Multi-task model with proposed method	iter.	EXPR	AU	*VA*	*Overall-M*
Simple born-again multi-task self-training	2	0.716	0.430	0.463	1.609
Simple born-again multi-task self-training	3	**0.729**	0.414	0.468	1.611
Simple born-again multi-task self-training	4	0.727	0.415	0.466	1.608
Task-specific born-again multi-task self-training	2	0.720	**0.444**	0.465	1.629
Task-specific born-again multi-task self-training	3	**0.729**	0.440	**0.472**	**1.641**
Task-specific born-again multi-task self-training	4	0.726	0.440	0.468	1.634

The results show the multi-task model with the conventional multi-task self-training achieved comparable emotion recognition performance to a single-task model with task-specific self-training. This indicates that the multi-task model has enough potential to improve emotion recognition performance while reducing inference cost compared with using multiple single-task models. However, this also indicates that conventional multi-task self-training did not yield synergy between tasks. The proposed simple born-again multi-task self-training yielded no performance improvements compared with conventional multi-task self-training. This suggests that pseudo-targets simply regenerated by using a multi-task model are not effective to improve the multi-task model. On the other hand, the proposed task-specific born-again multi-task self-training yielded performance improvements in each task. In addition, by increasing iterative born-again steps, the task-specific born-again self-training yielded further performance improvements. The highest performance was attained by task-specific born-again multi-task self-training with two born-again steps. These results suggest that task-specific born-again multi-task self-training is effective to train a multi-task model from multiple single-target annotated datasets and yield synergy between tasks. We consider that the synergy is attained when there is no inconsistency

Table 2. Evaluation of pseudo-targets in each born-again steps

Multi-task model with conventional method	iter.	EXPR	AU	*VA*
Multi-task self-training [11]	1	0.715	0.413	0.442
Multi-task model with proposed method	iter.	EXPR	AU	*VA*
Simple born-again multi-task self-training	2	0.719	0.430	0.462
Simple born-again multi-task self-training	3	0.716	0.430	0.463
Simple born-again multi-task self-training	4	0.729	0.414	0.468
Task-specific born-again multi-task self-training	2	0.735	0.443	0.476
Task-specific born-again multi-task self-training	3	0.738	**0.474**	**0.478**
Task-specific born-again multi-task self-training	4	**0.739**	**0.474**	0.474

between pseudo-targets for missing tasks and manually-annotated ground-truth targets. Note that iterative self-training steps are computationally expensive compared with task-specific training or conventional multi-task self-training. But, computation complexity in an inference step is exactly comparable with other training method.

Furthermore, we analyzed the performance of pseudo-targets in each born-again steps. We evaluated intermediate models that generate pseudo-targets. For conventional multi-task self-training, models via task-specific training was the intermediate ones. For simple born-again multi-task self-training, a multi-task model trained in a previous iteration step is the intermediate one. For task-specific born-again multi-task self-training, temporal single-task models trained in the process of the task-specific born-again multi-task self-training are the intermediate ones. We performed same evaluations for the EXPR, the AU, and VA tasks with Table 1. Table 2 shows the experimental results. The results show the intermediate models for task-specific born-again multi-task training outperformed those for conventional multi-task self-training and those for simple born-again multi-task self-training. This is because the fine-tuned temporal single-task models can utilize the good parts of multi-task knowledge through multi-task learning and task-specific knowledge at the same time. These results show that performance improvements by the proposed method were due to improvements of pseudo-targets. They also show that that the multi-task model was still no match for well-designed single-task models. Therefore, our future work is to reach the performance of temporal single-task models by using a multi-task model.

7 Conclusion

This paper proposed a novel approach for multi-task facial emotion recognition that utilizes single-target annotated datasets. We introduced a born-again multi-task self-training method to refine the pseudo-targets generated by the conventional multi-task self-training method. Our proposed method constructs a multi-task model that reduces computation costs compared to the single-task model. Experimental results showed that our approach improves the synergy among tasks and achieves better performance than the conventional multi-task self-training method. In future work, we will examine similar experiments using state-of-the-art vision Transformer based backbone architecture.

References

1. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. J. Pers. Soc. Psychol. **17**(2), 124–129 (1971)
2. Li, S., Deng, W.: Deep facial expression recognition: a survey. IEEE Trans. Affect. Comput. **13**(3), 1195–1215 (2022)
3. Martínez, B., Valstar, M.F., Jiang, B., Pantic, M.: Automatic analysis of facial actions: a survey. IEEE Trans. Affect. Comput. **10**(3), 325–347 (2019)
4. Russell, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. **39**(6), 1161–1178 (1980)
5. Kollias, D., Zafeiriou, S.: Expression, affect, action unit recognition: Aff-Wild2, multi-task learning and ArcFace. In: The British Machine Vision Conference (BMVC), pp. 297 (2019)
6. Deng, D., Chen, Z., Shi, B.E.: Multitask emotion recognition with incomplete labels. In: IEEE International Conference on Automatic Face and Gesture Recognition (FG), pp. 592–599 (2020)
7. Vu, M.-T., Beurton-Aimar, M., Marchand, S.: Multitask multi-database emotion recognition. In: IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3630–3637 (2021)
8. Kollias, D., Sharmanska, V., Zafeiriou, S.: Distribution Matching for Heterogeneous Multi-task Learning: a Large-Scale Face Study (2021). arXiv preprint: arXiv:2105.03790
9. Wang, L., Wang, S., Qi, J., Suzuki, K.: A multi-task mean teacher for semi-supervised facial affective behavior analysis. In: IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 3596–3601 (2021)
10. Thinh, P.T.D., Hung, H.M., Yang, H.-J., Kim, S.-H., Lee, G.-S.: Emotion recognition with sequential multi-task learning technique. In: IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), pp. 3586–3589 (2021)
11. Ghiasi, G., Zoph, B., Cubuk, E.D., Le, Q.V., Lin, T.-Y.: Multi-task self-training for learning general representations. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8836–8845 (2021)
12. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: International Conference on Machine Learning (ICML), vol. 80, pp. 1602–1611 (2018)
13. Lee, D.-H.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning (ICMLW), vol. 3 (2013)
14. Xie, Q., Luong, M.-T., Hovy, E.H., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695 (2020)
15. Goodfellow, I.J., et al.: Challenges in representation learning: a report on three machine learning contests. Neural Netw. **64**, 59–63 (2015)
16. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Recognition in the Wild, pp. 2584–2593 (2017)

17. Mavadati, S.M., Mahoor, M.H., Bartlett, K., Trinh, P., Cohn, J.F.: DISFA: a spontaneous facial action intensity database. IEEE Trans. Affect. Comput. **4**(2), 151–160 (2013)
18. Mollahosseini, A., Hassani, B., Mahoor, M.H.: AffectNet: a database for facial expression, valence, and arousal computing in the wild. IEEE Trans. Affect. Comput. **10**(1), 18–31 (2017)
19. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). In: IEEE International Conference on Computer Vision (ICCV), pp. 1021–1030 (2017)
20. Howard, A., et al.: Searching for MobileNetV3. In: International Conference on Computer Vision (ICCV), pp. 1314–1324 (2019)
21. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: a dataset for recognising face across pose and age. In: IEEE International Conference on Automatic Face & Gesture Recognition (FG), pp. 67–74 (2018)
22. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
23. Shrout, P.E., Fleiss, J.L.: Intraclass correlations: uses in assessing rater reliability. Psychol. Bull. **86**(2), 420–428 (1979)
24. Lawrence I-Kuei Lin: A concordance correlation coefficient to evaluate reproducibility. Biometrics **45**(1), 255–268 (1989)

Boosting Fine-Grained Oriented Object Detection via Text Features

Beichen Zhou, Qi Bi, Jian Ding, and Gui-Song Xia(✉)

Wuhan University, Wuhan, China
{beichen.zhou,q_bi,jian.ding,guisong.xia}@whu.edu.cn

Abstract. Fine-Grained Oriented Object Detection (FGOOD) aims to simultaneously categorize and localize fine-grained objects using oriented bounding box predictions. In this paper, we propose to exploit rich text features to discern fine-grained object categories from subordinate coarse-grained semantic classes, such as Boeing 747 vs. airplane. To this end, we leverage the emerging Contrastive Language-Image Pre-training (CLIP) model, which provides image-text representations to bridge the gap between oriented localization representations and fine-grained semantics. Our method is distinct from early FGOOD approaches which commonly focus on region proposal refinement but overlook the inter-class relations between fine-grained categories, resulting in inadequately discriminative features to discern the fine-grained categories. Specifically, our simple yet effective language-guided fine-grained oriented object detector first integrates hierarchical information from multi-granularity labels into a rotated object detection framework, establishing a shared representation space for Region of Interest (RoI) features and text features. Then, we extract fine-grained discriminative features from those RoI features using our elaborated Fine-grained Orthogonal Decomposition (FOD) and Fine-grained Orthogonal Feature Queue (FOQ). Extensive experiments validate the superiority of our approach, demonstrating a substantial performance improvement over state-of-the-art oriented object detectors on two FGOOD datasets, FAIR1M and HRSC2016, with a notable 1.67% and 3.42% mAP improvement on FAIR1M and HRSC2016.

Keywords: Oriented Object Detection · Fine-Grained · CLIP

1 Introduction

In recent years, the rapid development of oriented object detection [8,13,14,18,24,38,42] has been driven by advancements in deep learning and Earth Vision. Unlike object detection in generic images, which predicts Horizontal Bounding Boxes (HBBs) [1,21,29,32], oriented object detection focuses on detecting

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_8.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 109–125, 2025.
https://doi.org/10.1007/978-3-031-78444-6_8

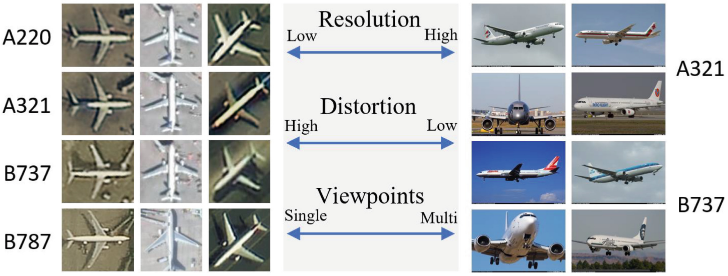


Fig. 1. Comparison between the FGOOD dataset and a conventional fine-grained classification dataset. In contrast to the right image from a conventional fine-grained classification dataset, FGVC-Aircraft [25], the left image from the FGOOD FAIR1M dataset [31] presents challenges such as low resolution, high distortion, and limited samples from highly varied viewpoints

objects in aerial images using Oriented Bounding Boxes (OBBs). Despite significant progress in accurately localizing OBBs, most state-of-the-art methods classify aerial objects only at a coarse granularity, such as distinguishing an airplane from a ship. However, practical Earth Vision applications require more fine-grained classification of aerial objects [31], such as differentiating between a Boeing 747 and an Airbus 350 within the airplane category.

This problem is known as Fine-Grained Oriented Object Detection (FGOOD), which aims to simultaneously localize and categorize fine-grained aerial objects using OBB predictions. Despite the challenges inherited from oriented object detection, such as arbitrary orientations and densely distributed small objects [36], FGOOD is particularly challenging in discerning specific fine-grained categories from others. As depicted in Fig. 1, aerial images commonly present unique challenges due to low resolution, high distortion, and limited samples from highly varied viewpoints, which deviate from the near-horizontal views prevalent in most natural images. As a result, aerial images often exhibit small inter-class variations and significant intra-class differences, making it extremely difficult to discern fine-grained categories.

Early FGOOD approaches primarily focus on improving oriented bounding box predictions from the detection head [3,6,16,19,26,27,30,33,44], but overlook the unique challenge of discerning fine-grained categories. As a result, the feature representation from these oriented object detectors remains insufficient for fine-grained categorization, leading to suboptimal performance of the classification head. In this paper, we focus on enhancing the oriented object representation to better discern fine-grained categories. We draw inspiration from the fine-grained visual categorization (FGVC) community, which also considers hierarchical relationships between categories [2,4,5,22,37].

Fortunately, the rich hierarchical semantic contexts for fine-grained classification can be leveraged by the emerging paradigm of Contrastive Language-Image Pre-training (CLIP) [28]. CLIP provides a natural way to harness connections between semantic categories of varying granularity. After large-scale pre-training on noisy image-text pairs, the text encoder of the CLIP model can map text into a vision-language common feature space, where the feature distributions of fine-grained classes within the same coarse class are close to each other and aligned with visual features. Motivated by this, we introduce multi-granularity text features from the text encoder of CLIP and employ Text Embedding Projection (TEP) learning to facilitate the alignment of region of interest (RoI) features with text features in a shared representation space.

Nevertheless, how to incorporate the hierarchical text representation between fine- and coarse-grained levels into the oriented detector representation remains an open question. To address this, we propose Fine-grained Orthogonal Decomposition (FOD) learning, aided by a Fine-grained Orthogonal Feature Queue (FOQ). Specifically, we orthogonally decompose the Rotated RoI (RRoI) features using the coarse-grained text features mapped by the TEP module. The resulting feature vectors are orthogonal to the discriminator vector of the coarse-grained category within the common feature space. This segment of the feature can be considered as the component that allows each fine-grained category within the same coarse category to distinguish itself from others [22,35,40]. We then apply supervised contrastive learning [17] to the fine-grained orthogonal features, encouraging the model to capture intricate and semantically rich features while mitigating the small inter-class variation problem.

Our contributions can be summarized as follows:

- We propose a language-guided fine-grained oriented object detector, dubbed LOOD, for FGOOD. *To the best of our knowledge*, this is the first work in the field to exploit vision-language models for representing fine-grained categories.
- We introduce a novel text embedding projection and fine-grained orthogonal decomposition learning to incorporate hierarchical fine-grained text representation into oriented detector representation.
- Comprehensive experiments demonstrate that the proposed method outperforms existing state-of-the-art methods by 1.67% and 3.42% mAP on the FAIR1M and HRSC2016 datasets, respectively.

2 Related Work

2.1 Vision-Language Models

(VLM) ingest data from both language and image modalities, and have drawn increasing attention in the past few years. A typical VLM paradigm for the vision community is contrastive language-image pre-training (CLIP) [28], which has been adopted in numerous downstream tasks. For example, ViLD [12] distills the CLIP knowledge to identify novel classes. DetPro [10] introduces the automatic prompt learning paradigm [46]. RegionCLIP is proposed to [45] to aid region-level classification inference. ZegFormer [9] is proposed to boost the segmentation ability for unseen (novel) classes. However, *to the best of our knowledge*, none of these works have leveraged CLIP to represent the fine-grained aerial images.

Oriented Object Detection employs Oriented Bounding Boxes (OBBs) to accommodate the arbitrary orientations of objects in aerial imagery. These methods can be classified into two categories, namely, one-stage [13] and two-stage [7,14,38]. More recently, Oriented RepPoints [18] has introduced an adaptive point learning approach. SASM [15] has improved the label assignment strategy. However, these methods are only designed for coarse-grained detection.

Fine-Grained Oriented Object Detection is predominantly evolved from oriented object detection. Some typical works include PCLDet [27], SFRNet [6], CF-ORNet [33] , RB-FPN [30] and etc. However, these methods usually focus on enhancing the feature representation for oriented bounding box predictions, but pay less attention to the relation between fine-grained categories. For FGOOD, leveraging the relation between fine- and coarse-grained categories can be critical to improving the category representation.

Fine-Grained Visual Categorization (FGVC) on natural images has undergone extensive research. Although the part-driven paradigm is the most common approach in this research direction [34], in recent years, more and more methods have begun to focus on the classification knowledge within multi-granularity labels [4,11,22,43]. These approaches leverage hierarchical label information to assist in training. Inspired by this trend, we introduce multi-granularity label knowledge into the FGOOD task.

3 Methodology

3.1 Revisiting Two-Stage Oriented Object Detector

Existing two-stage oriented object detectors (*e.g.,*, RoI Transformer, ReDet) typically follow the Cascade R-CNN paradigm. For an input image $\mathcal{I}$, the detection backbone processes it through the backbone and Feature Pyramid Network (FPN), generating multi-level convolutional features denoted as $f_{\mathcal{I}}$. These features are then fed into the Region Proposal Network (RPN), which produces horizontal region proposals emphasizing potential object-containing areas. RoI Align is applied to align features within these proposals to a fixed-size map, resulting in horizontal RoI features, denoted as f_{HRoI}. The first stage transforms HRoIs f_{HRoI} into RRoIs f_{RRoI}. Subsequently, a secondary RoI Align is performed based on the RRoI locations, generating rotated RoI features. The second stage f_{RRoI} undergoes another cycle of classification and regression, predicting rotated bounding boxes.

3.2 Problem Setup and Framework Overview

Fine-grained labels adhere to a hierarchical taxonomy structure. To align with dataset labels, we designate the most detailed category as the fine-grained category $\{l_F\}$. Coarser labels, one level above $\{l_F\}$, are defined as coarse-grained

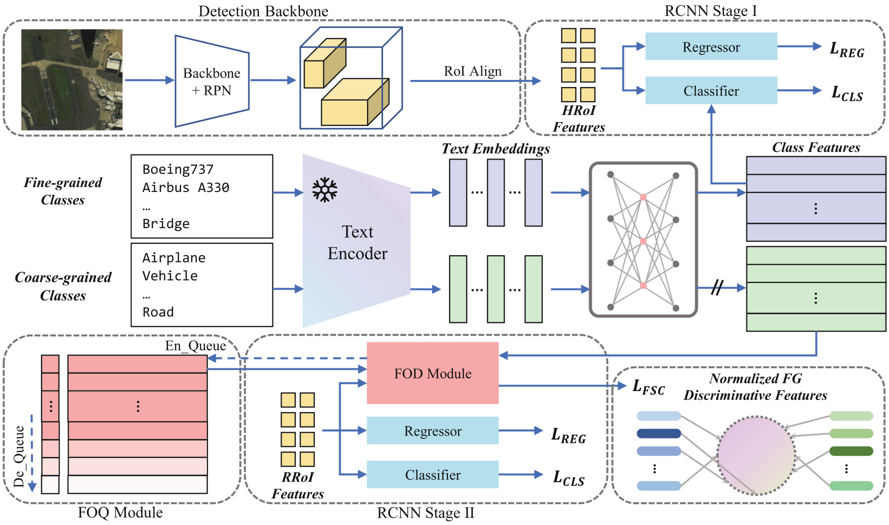


Fig. 2. Framework overview of the proposed language-guided fine-grained oriented object detector (LOOD) for fine-grained oriented object detection. The framework includes a novel text embedding projection learning (TEP, in Sect. 3.3) and fine-grained orthogonal decomposition learning (FOD, in Sect. 3.4) to incorporate hierarchical fine-grained text representation into the oriented detector representation. FOD is further supported by a Fine-Grained Orthogonal Feature Queue (FOQ, in Sect. 3.5)

categories $\{l_C\}$, given by

$$\begin{aligned}\{l_F\} &= \{l_{F,1}, l_{F,2}, ..., l_{F,|l_F|}\}\\ \{l_C\} &= \{l_{C,1}, l_{C,2}, ..., l_{C,|l_C|}\}\end{aligned} \tag{1}$$

where $l_{F,i}$ and $l_{C,i}$ denote the natural language names of the categories.

Given an input remote sensing image denoted as I, the objective of the FGOOD detector Φ is to identify and precisely locate fine-grained objects of interest within I using OBBs. This task can be denoted as $\Phi : I \rightarrow (x, y, w, h, \theta, l_{F,i})$, where x, y, w, h, θ denote the upper-left x-coordinate, upper-left y-coordinate, width, height, and long-edge angle of OBBs, respectively. $l_{F,i}$ denotes the fine-grained category.

Our LOOD is built on the RoI Transformer structure, as depicted in Fig. 2. After the detection backbone and the Region Proposal Network (RPN), our method comprises three key components: TEP, FOD, and FOQ. The TEP projects text embeddings of various granularities into a shared representation space synchronized with RoI features. Subsequently, the RRoI features are fed into the FOD module, which performs classification and regression. Fine-grained discriminative features are extracted through orthogonal decomposition in the common representation space. To enable the model to capture subtle differences between finer-grained details, we utilize the contrastive loss L_{FSC} on fine-grained discriminative features to reduce intra-class distance and increase inter-class distance. Additionally, we employ the FOQ module to interact with FOD, collecting high-quality fine-grained discriminative features for contrastive learning across multiple batches. FOQ enhances the optimization performance of L_{FSC} by increasing the number of positive and negative samples.

3.3 Text Embedding Projection

We leverage the CLIP model to generate text embeddings as a multi-grained representation. Due to CLIP's extensive vision-language pre-training, text embeddings can encapsulate the taxonomic knowledge within the multi-grained labels. We introduce Text Embedding Projection (TEP) learning to integrate this knowledge from text into the oriented detector. TEP learns a common representation space for RoI features and multi-grained class features through a projection learner. This module incorporates taxonomic knowledge and inter-class relations from text embeddings while accurately representing text features of coarse-grained classes in the joint feature space, facilitating feature decomposition in the subsequent Feature Orthogonal Decomposition (FOD) and Feature Orthogonal Query (FOQ) modules.

We combine $\{l_{CG}\}$ and $\{l_{FG}\}$, both with prompt templates (e.g., "a photo of a category"), and feed them into the CLIP text encoder $T(\cdot)$. Following the practices of ViLD [12], we ensemble multiple prompt templates to generate fine and coarse-grained text embeddings, denoted as e_{FG} and e_{CG}. To ensure the integrity of the original CLIP text embeddings during detector training, the parameters of $T(\cdot)$ are frozen.

We use a projection learner to map class features into the common feature space. The projection learner consists of lightweight bottleneck linear layers with a linear-ReLU-linear structure. e_{FG} and e_{CG} pass through the projection learner, denoted as

$$\begin{aligned} t_{FG} &= \mathrm{ReLU}(e_{FG}W_1 + B_1)W_2 + B_2 \\ t_{CG} &= \mathrm{ReLU}(e_{CG}W_1 + B_1)W_2 + B_2 \end{aligned} \tag{2}$$

where W_i and B_i denote the weight and bias of the fully connected layer, respectively.

Then, fine-grained category features t_{FG} act as classification weights for RCNN Stage One. To align HRoI features with t_{FG}, we modify the RCNN Stage One classifier to a cosine similarity classifier akin to CLIP. The classification logits for the ith RoI and jth class are computed as the scaled cosine similarity between HRoI features and fine-grained text features in the hypersphere:

$$\mathrm{Logit}_{i,j} = \lambda \cdot \frac{f_{\mathrm{HRoI}_i} \cdot t_{\mathrm{FG}_j}}{\|f_{\mathrm{HRoI}_i}\| \cdot \|t_{\mathrm{FG}_j}\|} \tag{3}$$

where λ is a scaling factor, and $\| \cdot \|$ denotes the Euclidean norm. The modified classifier employs conventional cross-entropy loss to optimize the classification

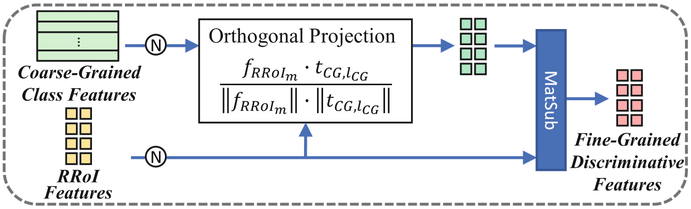

Coarse-Grained Class Features
N
Orthogonal Projection
$\frac{f_{RRoI_m} \cdot t_{CG,l_{CG}}}{\|f_{RRoI_m}\| \cdot \|t_{CG,l_{CG}}\|}$
MatSub
RRoI Features
N
Fine-Grained Discriminative Features

Fig. 3. Illustration of the proposed fine-grained orthogonal decomposition learning (FOD)

process:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{|l_F|} y_{i,j} \cdot \log(p_{i,j}) \tag{4}$$

where N denotes the number of RoIs, and $p_{i,j}$ is the softmax probability of $\text{Logit}_{i,j}$.

During backpropagation, the projection learner is optimized. By minimizing the classification loss, the projection learner is expected to establish a bridge from the original vision-language feature space of CLIP to the RoI features in remote sensing images. Note that, although t_{CG} plays a crucial role in the subsequent module training processes, the gradients of the projection learner originate solely from the classification process in RCNN Stage One.

3.4 Fine-Grained Orthogonal Decomposition

Conventional RoI features may suffice for coarse-grained classification tasks, but they often fall short of capturing subtle differences for fine-grained classification. To address this limitation, we introduce Fine-grained Orthogonal Decomposition (FOD) learning. Motivated by [22,35,40], this module aims to disentangle fine-grained discriminative features from coarse-grained features.

FOD operates concurrently with the classification and regression branches in RCNN Stage Two, as depicted in Fig. 3. Initially, both f_{RRoI} and coarse-grained class features t_{CG} undergo Euclidean normalization. Subsequently, we compute the projection of the m-th RRoI feature onto its corresponding coarse-grained class feature $t_{CG,l_{CG}}$:

$$\text{Proj}(f_{RRoI_m}, t_{CG,l_{CG}}) = \frac{f_{RRoI_m} \cdot t_{CG,l_{CG}}}{\|f_{RRoI_m}\| \cdot \|t_{CG,l_{CG}}\|} \tag{5}$$

where l_{CG} denotes the coarse-grained category of the ground truth matching the m-th RRoI. In the joint feature space constructed by the TEP module, this projection vector aligns with the coarse-grained class feature, encapsulating common features of the respective coarse-grained category.

The representation of fine-grained discriminative features characterizing the RRoI is obtained by calculating the orthogonal component of the RRoI feature:

$$\text{Ortho}(f_{RRoI_m}, t_{CG,l_{CG}}) = f_{RRoI_m} - \text{Proj}(f_{RRoI_m}, t_{CG,l_{CG}}) \tag{6}$$

In RCNN Stage One, through the optimization of the classification loss, we align the HRoI features with fine-grained class features. Therefore, on the hypersphere with a radius of 1, the m-th HRoI feature is nearly in the same direction as its corresponding fine-grained class features. However, after the regression correction in RCNN Stage One and the RRoI Align process, as shown by the yellow vector in Fig. 3, f_{RRoI} is distributed in the vicinity of its fine-grained class feature. We calculate the orthogonal component of f_{RRoI} with respect to its coarse-grained class feature. This set of features characterizes finer-grained, unique details and is a crucial component for distinguishing different fine-grained subclasses within the same coarse-grained category.

3.5 Fine-Grained Orthogonal Feature Queue

We employ a contrastive loss on the extracted fine-grained discriminative features to minimize intra-class distances and maximize inter-class distances. The efficacy of contrastive learning relies on the quantity and quality of positive and negative samples. To enhance this, we introduce a Fine-grained Orthogonal Feature Queue (FOQ) responsible for storing high-quality feature samples. Throughout the detector training, FOQ collaborates with the FOD module, updating the queue, and the stored samples contribute to subsequent contrastive loss calculations.

We consider only positive RRoIs matching the ground truth (GT) and use Intersection over Union (IoU) with the GT RBBox as a metric for assessing the quality of fine-grained discriminative features. Let $\mathcal{Q}$ represent FOQ, which we express as:

$$\mathcal{Q} = [(\mathbf{v}_1, c_1), (\mathbf{v}_2, c_2), \ldots, (\mathbf{v}_l, c_l)] \tag{7}$$

Here, $\mathbf{v}_i$ denotes the feature vector from FOD, and c_i denotes the corresponding fine-grained class of the RRoI. The length of the list is denoted as l.

During training, $\mathbf{v}_i$ is randomly initialized, and c_i is initialized to $|l_F| + 1$, representing the background class in the detector. Filtering samples with an IoU greater than 0.5, we enqueue their fine-grained discriminative features:

$$\mathcal{Q} \leftarrow \{(\mathbf{v}_i, c_i) \mid \text{IoU}(\text{GT RBBox}, \text{RRoI}_i) > 0.5\} \tag{8}$$

Given that gradient backpropagation occurs in each iteration, outdated sample features should not persist in training. To maintain the queue's timeliness, samples at the tail will be dequeued as new samples are added.

After updating the queue, we compute the contrastive loss for fine-grained discriminative features. Following the approach in [17], we calculate the Fine-grained Supervised Contrastive Loss (L_{FSC}) for high-quality FOQ samples and the samples generated in the current mini-batch. To ensure sample quality, only those with an IoU greater than 0.4 from the current mini-batch participate in the L_{FSC} calculation. Moreover, only higher-quality samples with an IoU exceeding 0.5 are added to FOQ. Let $\mathcal{F}$ denote the full set of features involved in the L_{FSC} computation. We have

$$\begin{aligned}
&\mathcal{F} = \mathcal{F}_{\text{minibatch}} \cup \mathcal{F}_{\text{FOQ}}, \\
&\mathcal{F}_{\text{minibatch}} = \{(\mathbf{v}_i, c_i) \mid \text{IoU}(\text{GT RBBox}, \text{RBBox}_i) > 0.4\}, \\
&\mathcal{F}_{\text{FOQ}} = \{(\mathbf{v}_i, c_i) \mid c_i \neq (|l_F| + 1)\}.
\end{aligned} \tag{9}$$

The computation process of L_{FSC} is expressed as:

$$\begin{aligned}
&\mathcal{L}_{FSC}(\mathcal{F}) = -\frac{1}{|\mathcal{F}|} \sum_{(\mathbf{v}_i, c_i) \in \mathcal{F}} \\
&\quad \log \left(\frac{\exp(\text{sim}(f(\mathbf{v}_i), f(x_j))/\tau)}{\sum_{(\mathbf{v}_k, c_k) \in \mathcal{F}} \mathbb{1}_{[c_k \neq c_i]} \exp(\text{sim}(f(\mathbf{v}_i), f(\mathbf{v}_k))/\tau} \right).
\end{aligned} \tag{10}$$

where

$$\mathbb{1}_{[c_k \neq c_i]} = \begin{cases} 1 & \text{if } c_k \neq c_i, \\ 0 & \text{otherwise} \end{cases}$$

3.6 Training Objective

We denote the classification and regression losses in the conventional RoI Transformer's two stages as L_{cls} and L_{reg}, respectively. The loss generated by the RPN part is denoted as L_{RPN}. The training objective of LOOD is formulated as

$$L = L_{RPN} + L_{cls} + L_{reg} + \alpha L_{FSC},$$

where L_{FSC} follows the formulation in Eq. 9. Here, α serves as a hyperparameter, typically set to 0.5. Its purpose is to normalize the values of each loss to the same order of magnitude, preventing the training of the detector from being dominated by a single loss.

4 Experiments

4.1 Datasets

FAIR1M [31] is the largest fine-grained aerial object detection dataset to date, comprising 42,762 high-resolution images and over a million annotated objects across 5 categories and 37 sub-categories using oriented bounding boxes. We utilized the FAIR1M-2.0 training set and evaluated on its validation subset due to the unavailability of the test set.

HRSC2016 [23] is a specialized dataset for ship detection, with 1,061 images ranging from 0.4m to 2.0m resolution. It includes three levels of category labels, providing both category-level and type-level annotations across 27 fine-grained categories. We used the second and third-level annotations as coarse-grained and fine-grained categories, respectively, and reported accuracy based on the test set.

4.2 Implementation Details

As our proposed LOOD serves as the RoI Head of a rotation detector, the standard version of LOOD based on RoI Transformer is denoted as LOOD (RT) and employs ResNet50 and Feature Pyramid Network as the detection backbone. We also implement LOOD integrated with ReDet, denoted as LOOD (RD). LOOD (RD) uses ReResNet50 and ReFPN as the detection backbone, and the second RoI Align employs the RiRoIAlign algorithm.

To maintain the stability of samples in the Feature Orthogonal Queue (FOQ), we use the Exponential Moving Average algorithm to update the model parameters. For a fair comparison, we also adopt this algorithm during the training of other methods. All detectors are trained using ResNet50 pretrained on ImageNet. The SGD optimizer is employed during training, with an initial learning rate set to 5×10^{-3}, momentum and weight decay set to 0.9 and 1×10^{-4}, respectively. Specifically, to maintain the stability of the common feature space in the TEP module, the learning rate of the projection learner in the TEP module is separately set to 1×10^{-4}. Training lasts for 12 epochs on the FAIR1M dataset and 36 epochs on HRSC2016.

4.3 Comparison with State-of-the-Art

Results on FAIR1M. The experimental results on the FAIR1M dataset are presented in Table 1. Due to space constraints, we have placed the table for ship categories in the Supplementary Materials. We compared our proposed LOOD with well-known oriented object detectors. LOOD (RT) and LOOD (RD) achieved mAP scores of 42.30 and 44.90, respectively. Notably, LOOD (RD) outperformed all other detectors, while LOOD (RT) surpassed all detectors using ResNet50. Compared to the baseline RoI Transformer and baseline ReDet, LOOD demonstrated improvements of 2.09 and 1.67 points, respectively.

Among the total of 37 fine-grained categories, LOOD (RD) excelled in 21 categories. In the challenging Airplane category, where fine-grained models are harder to distinguish, LOOD (RD) achieved the best performance in 9 out of 11 fine-grained models. In categories like court and road, where fine-grained attributes are less distinctive (i.e., significant differences between fine-grained classes), LOOD and baseline versions of detectors showed mixed results. It can be concluded that LOOD provides significant performance improvement, particularly in challenging and less distinguishable fine-grained categories. For those fine-grained categories that can be easily distinguished, opting not to employ LOOD for contrastive learning is a more favorable choice.

Table 1. Results comparison between the proposed LOOD and existing oriented object detectors. Experiments conducted on FAIR1M Dataset. The evaluation metric $mAP50$ is presented in percentage

Coarse Cat.	Fine Cat.	SASM RepPoints [15]	R-FCOS [20]	S2A-Net [13]	R-Faster RCNN [41]	O-Rep-Points [18]	O-RCNN [38]	RoI Trans [7]	LOOD (RT)	ReDet [14]	LOOD (RD)
		AAAI 2022	ICCV 2019	TGRS 2021	PAMI 2017	CVPR 2022	ICCV 2021	CVPR 2019	-	CVPR 2021	-
Airplane	Boeing737	38.3	39.2	40.4	37.5	37.7	37.6	38.9	**44.6**	37.1	43.1
	Boeing747	61.7	82.7	83.7	82.5	81.4	84.8	81.8	83.4	86.6	**90.6**
	Boeing777	15.0	15.9	17.2	15.4	15.4	16.9	14.6	18.5	16.5	**22.2**
	Boeing787	35.4	52.0	49.9	47.7	48.8	49.0	46.1	47.9	**57.2**	55.8
	C919	0.4	0.2	0.2	9.4	7.9	7.3	7.1	**11.7**	9.4	9.4
	A220	44.0	41.0	42.5	41.7	43.4	41.6	42.4	44.6	44.9	**46.6**
	A321	53.3	54.5	57.0	50.8	54.6	49.7	55.4	59.4	59.1	**63.7**
	A330	30.6	44.2	42.7	47.5	43.5	43.1	38.6	47.5	45.4	**55.8**
	A350	20.7	43.5	52.0	60.6	56.6	62.4	57.3	59.9	60.3	**68.3**
	ARJ21	3.9	9.3	10.3	8.3	9.5	13.1	10.3	**18.0**	11.1	16.5
	Other Airplane	70.7	73.1	74.8	72.5	73.3	72.1	73.4	76.1	76.4	**77.0**
Vehicle	Small Car	53.6	55.1	**66.2**	57.7	62.9	59.8	62.2	63.3	63.7	64.9
	Bus	9.4	11.5	13.4	12.9	15.4	18.5	20.6	20.24	20.9	**23.3**
	Cargo Truck	29.9	37.9	39.7	41.4	42.0	43.9	43.7	45.0	44.2	**45.8**
	Dump Truck	21.2	26.0	35.7	38.2	42.8	41.5	43.4	**45.5**	42.7	**45.5**
	Van	48.3	51.0	**61.7**	53.4	59.5	56.0	58.5	59.1	59.7	61.2
	Trailer	2.8	6.3	2.8	11.7	4.5	7.9	**13.7**	10.4	13.2	12.5
	Tractor	0.1	1.1	0.8	2.1	0.6	2.2	1.9	2.2	1.8	**4.8**
	Excavator	2.2	12.9	10.5	16.8	17.0	**24.4**	21.0	21.7	24.2	19.0
	Truck Tractor	0.1	5.4	2.5	11.9	5.7	10.4	16.5	**22.2**	16.7	14.9
	Other Vehicle	2.4	**2.7**	2.2	2.0	0.9	1.2	1.3	1.7	1.4	1.6
Court	Basketball Court	37.2	43.7	46.2	42.6	46.3	47.2	50.4	52.31	**55.8**	54.2
	Tennis Court	80.4	86.2	82.1	83.3	84.3	83.3	**86.2**	84.7	**86.2**	84.6
	Football Field	49.9	56.3	59.0	55.4	61.4	59.6	61.0	**67.3**	66.8	64.7
	Baseball Field	87.4	88.1	88.2	89.4	89.2	89.1	88.7	89.8	91.0	**92.4**
Road	Intersection	44.5	48.0	41.5	50.4	46.7	48.3	50.2	50.8	**51.1**	50.7
	Roundabout	54.8	57.2	62.9	65.1	60.8	62.6	66.3	68.2	**74.5**	71.7
	Bridge	28.3	25.7	20.3	26.1	33.2	31.8	30.7	37.3	37.6	**41.2**
mAP		30.86	36.1	37.42	37.52	38.9	40.38	40.21	42.57	43.23	**44.90**

Table 2. Results comparison on HRSC2016

Methods	R-Faster RCNN [41]	R-FCOS [20]	Gliding Vertex [39]	S2A-Net [13]	Oriented RCNN [38]	RoI Trans [7]	LOOD (RT)	Redet [14]	LOOD (RD)
mAP	11.86	12.34	19.50	22.96	39.01	38.16	40.11	51.74	**55.16**

Results on HRSC2016. The experimental results on the HRSC2016 dataset are presented in Table 2. Due to the dataset's relatively small size and the high difficulty of fine-grained tasks, previous studies using this dataset have mostly employed single-category label experiments. In fine-grained tasks, LOOD (RD) outperforms all other detectors. Compared to the baseline RoI Transformer and baseline ReDet, LOOD achieves improvements of 2.09 and 1.67 points, respectively.

4.4 Ablation Studies

On Each Component. Table 3a systematically elucidates the impact of various components in our proposed LOOD. When employing only the TEP module-meaning the utilization of class features solely for classification in RCNN Stage One-the model's detection accuracy is scarcely affected. However, incorporating the FOD module, which decouples RoI features and employs contrastive learning, significantly enhances detection accuracy. In the FAIR1M dataset, this enhancement results in a 0.95% mAP increase. The inclusion of FOQ further improves the optimization effectiveness of L_{FSC}.

On Length of FOQ. Additionally, we conducted a series of experiments regarding the choice of FOQ length, as presented in Table 3b. We set the FOQ lengths to 256, 512, 1024, and 2048. The results indicate that at lower queue lengths (256 and 512), increasing the length enriches the number of positive and negative samples in L_{FSC}, leading to respective mAP increments of 0.94% and 1.36% compared to not using FOQ. However, when the queue length reaches 1024, the model's accuracy slightly decreases. We attribute this to excessively long queues retaining outdated features, which conflict with the current model's feature space and render L_{FSC} relatively inefficient.

Different Selection of Text Features. To investigate the impact of text embedding quality on LOOD, we present the mAP metrics using text embeddings from different sources on the FAIR1M dataset in Table 3c. Although the ViT-B/32 version demonstrates significantly stronger zero-shot capabilities compared to the RN50 version when generating text embeddings with various CLIP pre-trained models, their performance on LOOD does not show substantial differences. Word2Vec is a commonly used method in NLP for obtaining word embeddings, and text embeddings acquired using a pre-trained Word2Vec model can compute the similarity between two words. The experiments reveal that CLIP text embeddings outperform those from Word2Vec. These meaningful text

Table 3. Ablation Study Results. Experimental setting: LOOD on FAIR1M

(a) Ablation study on each component.

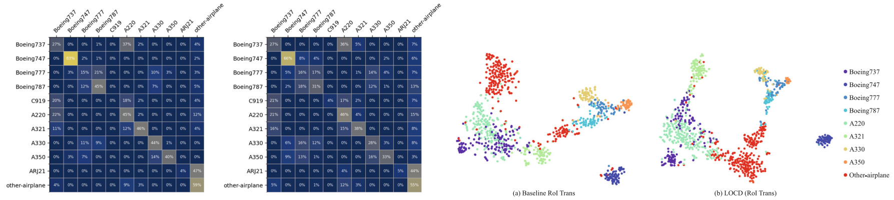
Component			mAP
TEP	FOD	FOQ	
			40.21
✓			40.26
✓	✓		41.21
✓	✓	✓	**42.57**

(b) Results with different FOQ lengths.

Length of FOQ	mAP
256	42.15
512	**42.57**
1024	42.34
2048	41.96

(c) mAP values for different sources of text embeddings.

Source of Text Embeddings	mAP
CLIP (RN50)	**42.57**
CLIP (ViT-B/32)	42.28
random init	41.10
Word2Vec	41.85



(a) Visualized confusion matrix. (b) t-SNE visualization on GT RoIs' classification features.

Fig. 4. (a) Visualized confusion matrix. (b) t-SNE visualization on GT RoIs' classification features embeddings, compared to randomly initialized vectors, contribute to enhancing the model's detection performance.

On Confusion Matrix. We present the confusion matrix for the coarse category "airplane" in the FAIR1M dataset, comparing LOOD (RT) with the baseline RoI Transformer, as shown in Fig. 4a. The values on the diagonal indicate that LOOD (RT) enhances category recognition performance in most fine-grained categories, notably in Boeing747 (83% vs. 66%), A321 (46% vs. 38%), and A330 (44% vs. 28%).

Feature Space Visualization by t-SNE. We conducted t-SNE visualization on the classification features of Ground Truth (GT) Regions of Interest (RoIs), as shown in Fig. 4b. The left image corresponds to the baseline RoI Transformer, while the right image represents the results of LOOD (RT). Since we visualize only the GT RoIs, this comparison is fair and unaffected by regression performance. The visualization reveals that LOOD (RT) exhibits a more compact feature distribution for most categories, particularly evident in the "other-airplane" category. However, due to the inherent challenges of the FGOOD task, as illustrated in Fig. 1, the classification boundaries for some categories, while improved compared to the baseline, remain less distinct (e.g., A220 and Boeing737). Additionally, some extremely challenging samples affected by distortions still lead to misclassifications. This observation aligns with the outcomes presented in our confusion matrix, shown in Fig. 4a.

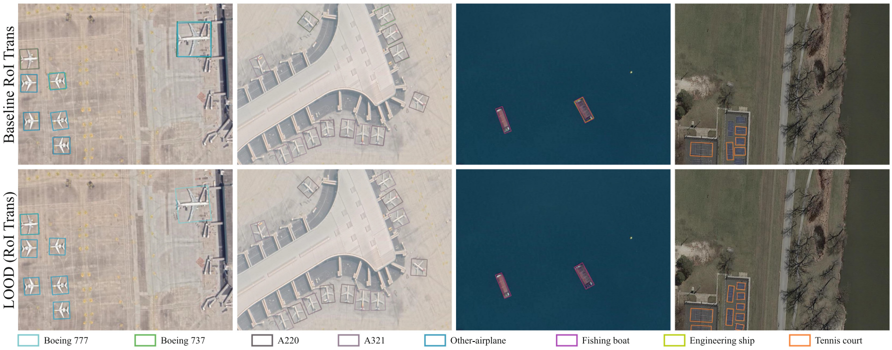


Fig. 5. Visual Detection Results. Zoom in for better view

4.5 Visual Detection Results

Our visualization of detection results is depicted in Fig. 5. The first row illustrates the detection results of the baseline RoI Transformer, while the second row presents the results of LOOD (RT). Comparing the detection results for the same image, LOOD shows a significant reduction in instances where multiple bounding boxes appear for the same target, highlighting an improvement in classification accuracy. Additionally, LOOD contributes to an enhanced detection recall rate for certain categories, such as Tennis Court.

5 Conclusion

In this paper, we propose to exploit rich text features for Fine-Grained Oriented Object Detection. By leveraging the Contrastive Language-Image Pre-training (CLIP) model, we present a straightforward yet effective method for detecting detailed-oriented objects. Our approach involves integrating information from different label levels into a detection framework, establishing a shared space for image and text features. Subsequently, we extract detailed features using our proposed Fine-grained Orthogonal Decomposition and Fine-grained Orthogonal Feature Queue modules. Our extensive experiments have validated the superiority of the proposed method. Our work demonstrates the potential of leveraging pretrained VLM to enhance closed-set tasks.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154–6162 (2018)
2. Chang, D., Pang, K., Zheng, Y., Ma, Z., Song, Y.Z., Guo, J.: Your “flamingo” is my “bird”: fine-grained, or not. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11476–11485 (2021)
3. Chen, D., Zhong, Y., Ma, A., Zheng, Z., Zhang, L.: Explicable fine-grained aircraft recognition via deep part parsing prior framework for high-resolution remote sensing imagery. IEEE Trans. Cybernet., (2023)
4. Chen, J., Wang, P., Liu, J., Qian, Y.: Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4858–4867 (2022)
5. Chen, T., Wu, W., Gao, Y., Dong, L., Luo, X., Lin, L.: Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 2023–2031 (2018)
6. Cheng, G., Li, Q., Wang, G., Xie, X., Min, L., Han, J.: Sfrnet: Fine-grained oriented object recognition via separate feature refinement. IEEE Trans. Geosci. Remote Sens., (2023)
7. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning roi transformer for oriented object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2849–2858 (2019)
8. Ding, J., et al.: Object detection in aerial images: a large-scale benchmark and challenges. IEEE Trans. Pattern Anal. Mach. Intell. **44**(11), 7778–7796 (2021)
9. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11583–11592 (2022)

10. Du, Y., Wei, F., Zhang, Z., Shi, M., Gao, Y., Li, G.: Learning to prompt for open-vocabulary object detection with vision-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14084–14093 (2022)
11. Garg, A., Sani, D., Anand, S.: Learning hierarchy aware features for reducing mistake severity. In: European Conference on Computer Vision, pp. 252–267. Springer (2022)
12. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-Vocabulary Object Detection via Vision and Language Knowledge Distillation (2021). arXiv preprint arXiv:2104.13921
13. Han, J., Ding, J., Li, J., Xia, G.S.: Align deep features for oriented object detection. IEEE Trans. Geosci. Remote Sens. **60**, 1–11 (2021)
14. Han, J., Ding, J., Xue, N., Xia, G.S.: Redet: a rotation-equivariant detector for aerial object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2786–2795 (2021)
15. Hou, L., Lu, K., Xue, J., Li, Y.: Shape-adaptive selection and measurement for oriented object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 923–932 (2022)
16. Huang, L., Wang, F., Zhang, Y., Xu, Q.: Fine-grained ship classification by combining cnn and swin transformer. Remote Sens. **14**(13), 3087 (2022)
17. Khosla, P., et al.: Supervised contrastive learning. Adv. Neural. Inf. Process. Syst. **33**, 18661–18673 (2020)
18. Li, W., Chen, Y., Hu, K., Zhu, J.: Oriented reppoints for aerial object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1829–1838 (2022)
19. Li, Y., Wang, Q., Luo, X., Yin, J.: Class-balanced contrastive learning for fine-grained airplane detection. IEEE Geosci. Remote Sens. Lett. **19**, 1–5 (2022)
20. Li, Z., Hou, B., Wu, Z., Jiao, L., Ren, B., Yang, C.: Fcosr: a Simple Anchor-Free Rotated Detector for Aerial Object Detection (2021). arXiv preprint arXiv:2111.10780
21. 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)
22. Liu, Y., e al.: Where to focus: investigating hierarchical attention relationship for fine-grained visual classification. In: European Conference on Computer Vision, pp. 57–73. Springer (2022)
23. Liu, Z., Wang, H., Weng, L., Yang, Y.: Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. IEEE Geosci. Remote Sens. Lett. **13**(8), 1074–1078 (2016)
24. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans. Multimed. **20**(11), 3111–3122 (2018)
25. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-Grained Visual Classification of Aircraft (2013). arXiv preprint arXiv:1306.5151
26. Ming, Q., Song, J., Zhou, Z.: Oriented Feature Alignment for Fine-Grained Object Recognition in High-Resolution Satellite Imagery (2021). arXiv preprint arXiv:2110.06628
27. Ouyang, L., Guo, G., Fang, L., Ghamisi, P., Yue, J.: Pcldet: Prototypical contrastive learning for fine-grained object detection in remote sensing images. IEEE Trans. Geosci. Remote Sens., (2023)

28. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst., **28** (2015)
30. Song, J., Miao, L., Ming, Q., Zhou, Z., Dong, Y.: Fine-grained object detection in remote sensing images via adaptive label assignment and refined-balanced feature pyramid network. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **16**, 71–82 (2022)
31. Sun, X., Wang, P., Yan, Z., Xu, F., Wang, R., Diao, W., Chen, J., Li, J., Feng, Y., Xu, T., et al.: Fair1m: a benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. ISPRS J. Photogramm. Remote. Sens. **184**, 116–130 (2022)
32. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
33. Wang, L., Zhang, J., Tian, J., Li, J., Zhuo, L., Tian, Q.: Efficient fine-grained object recognition in high-resolution remote sensing images from knowledge distillation to filter grafting. IEEE Trans. Geosci. Remote Sens. **61**, 1–16 (2023)
34. Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S.: Fine-grained image analysis with deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 8927–8948 (2021)
35. Wu, A., Liu, R., Han, Y., Zhu, L., Yang, Y.: Vector-decomposed disentanglement for domain-invariant object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9342–9351 (2021)
36. Xia, G.S., et al.: Dota: a large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3974–3983 (2018)
37. Xie, L., Tian, Q., Hong, R., Yan, S., Zhang, B.: Hierarchical part matching for fine-grained visual categorization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1641–1648 (2013)
38. Xie, X., Cheng, G., Wang, J., Yao, X., Han, J.: Oriented r-cnn for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3520–3529 (2021)
39. Xu, Y., Fu, M., Wang, Q., Wang, Y., Chen, K., Xia, G.S., Bai, X.: Gliding vertex on the horizontal bounding box for multi-oriented object detection. IEEE Trans. Pattern Anal. Mach. Intell. **43**(4), 1452–1459 (2020)
40. Yang, M., et al.: Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In: Proceedings of the IEEE/CVF International conference on Computer Vision, pp. 11772–11781 (2021)
41. Yang, S., Pei, Z., Zhou, F., Wang, G.: Rotated faster r-cnn for oriented object detection in aerial images. In: Proceedings of the 2020 3rd International Conference on Robot Systems and Applications, pp. 35–39 (2020)
42. Yang, X., Yan, J., Feng, Z., He, T.: R3det: Refined single-stage detector with feature refinement for rotating object. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 3163–3171 (2021)
43. Zeng, S., des Combes, R.T., Zhao, H.: Learning structured representations by embedding class hierarchy. In: The Eleventh International Conference on Learning Representations (2022)
44. Zhang, R., Xie, C., Deng, L.: A fine-grained object detection model for aerial images based on yolov5 deep neural network. Chin. J. Electron. **32**(1), 51–63 (2023)

45. Zhong, Y., et al.: Regionclip: region-based language-image pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16793–16803 (2022)
46. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vision **130**(9), 2337–2348 (2022)

Large-Scale Pre-trained Models are Surprisingly Strong in Incremental Novel Class Discovery

Mingxuan Liu[1], Subhankar Roy[3], Zhun Zhong[4](✉), Nicu Sebe[1], and Elisa Ricci[1,2]

[1] University of Trento, Trento, Italy
mingxuan.liu@unitn.it
[2] Fondazione Bruno Kessler, Trento, Italy
[3] University of Aberdeen, Aberdeen, UK
[4] University of Nottingham, Nottingham, UK
zhunzhong007@gmail.com

Abstract. Discovering novel concepts in unlabelled datasets and in a continuous manner is an important desideratum of lifelong learners. In the literature such problems have been partially addressed under very restricted settings, where novel classes are learned by jointly accessing a related labelled set (e.g., NCD) or by leveraging only a supervisedly pre-trained model (e.g., class-iNCD). In this work we challenge the status quo in class-iNCD and propose a learning paradigm where class discovery occurs continuously and truly unsupervisedly, without needing any related labelled set. In detail, we propose to exploit the richer priors from strong self-supervised pre-trained models (PTM). To this end, we propose simple baselines, composed of a frozen PTM backbone and a learnable linear classifier, that are not only simple to implement but also resilient under longer learning scenarios. We conduct extensive empirical evaluation on a multitude of benchmarks and show the effectiveness of our proposed baselines when compared with sophisticated state-of-the-art methods. The code is open source.

Keywords: Novel Class Discovery · Class-Incremental Learning

1 Introduction

Clustering unlabelled samples in a dataset is a long standing problem in computer vision, where the goal is to group samples into their respective semantic categories. Given, there could be multiple valid criteria (e.g., shape, size or color) that could be used to cluster data, Deep Clustering (DC) [48] can at times lead to clusters without desired semantics. A more efficient alternative was proposed

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_9.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 126–142, 2025.
https://doi.org/10.1007/978-3-031-78444-6_9

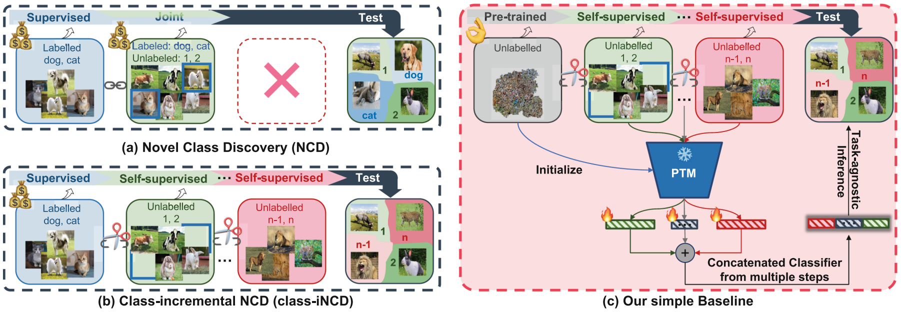


Fig. 1. Overview of different learning paradigms for discovering novel (or *new*) categories from *unlabelled* data. (a) **NCD** learns and discovers novel classes in an unalabelled dataset by exploiting the priors learned from related labelled data. (b) **class-iNCD** is similar to NCD, except it discovers novel classes arriving in sessions without any access to labelled data during the discovery phase. (c) Our proposed simple Baseline for class-iNCD that leverages a self-supervised pre-trained model (PTM) instead of expensive labelled data. Inference on test data is carried out in a task-*agnostic* manner.

in the work of Novel Class Discovery (NCD) [18], where the goal is to discover and learn new semantic categories in an unlabelled dataset by transferring prior knowledge from labelled samples of related yet disjoint classes (see Fig. 1a). In other words, NCD can be viewed as unsupervised clustering guided by known classes. Due to its practical usefulness, the field of NCD has seen a tremendous growth, with application areas ranging from object detection [16] to 3D point clouds [39].

A commonality in most of the NCD methods [15,19] is that they rely on a reasonably large labelled dataset to learn good categorical and domain priors about the dataset. Thus, the success of these methods rely entirely on the availability of large labelled datasets, which might not always be guaranteed or can be expensive to collect. In this work we challenge the *de facto* supervised pre-training step on a large labelled dataset for NCD and show that supervised pre-training can be easily replaced by leveraging self-supervised pre-trained models (PTM), such as DINO [6]. PTMs being readily available off-the-shelf, it reduces the burden of pre-training on labelled data. As a part of a preliminary study, we compare supervised pre-training with PTMs and analyse their impact on the novel classes performance. As shown in Fig. 2, the PTMs achieve significantly better or at-par performance in comparison with

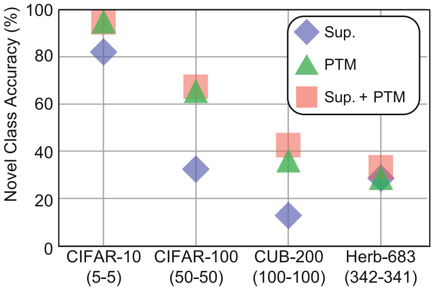


Fig. 2. Comparison of traditional **Supervised** pre-training (Sup.) with self-supervised **pre-trained model** (PTM) initialization on the Novel Class Discovery.

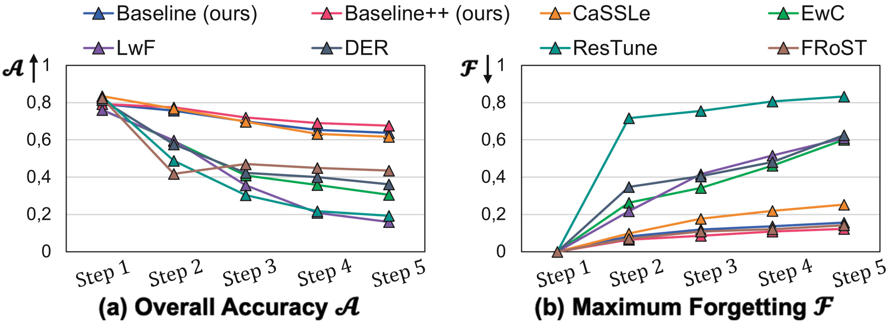


Fig. 3. Comparison of our proposed baselines with the incremental learning (EwC, LwF, DER), unsupervised incremental learning (CaSSLe), and iNCD (ResTune, FRoST) methods on CIFAR-100. In each step 20 novel classes are learned. We report the Overall Accuracy and Maximum Forgetting.

the only supervised counterparts on all the datasets. Furthermore, when the PTMs are fine-tuned with supervised training on the labelled data, the performance is only marginally better. Note that the work in GCD [43] used DINO as PTM, except it is used as initialization for the supervised training. Contrarily, we propose to entirely get rid of the supervised step.

Another striking drawback of the vast majority of NCD methods, especially in [15,19], is that they assume access to the labelled dataset while discovering the novel (or *new*) classes. Due to storage and privacy reasons the access to the labelled dataset can be revoked, which makes NCD a very challenging problem. To address this, some very recent Class-incremental Novel Class Discovery (class-iNCD) methods [26,40] have attempted to address NCD from the lens of continual learning, by not accessing the labelled dataset when learning new classes (see Fig. 1b). Albeit more practical than NCD, the class-iNCD methods are still susceptible to catastrophic forgetting [17], thereby impairing knowledge transfer from the labelled set to the unlabelled sets.

In this work we aim to create a simple yet strong baseline for class-iNCD that can continually learn to cluster unlabelled data arriving in sessions, without losing its ability to cluster previously seen data. To this end, we propose `Baseline` (see Fig. 1c) that uses the DINO pre-trained ViT backbone, as a *frozen* feature extractor, with a learnable linear *cosine normalized* classifier [21] on top. Every time an unlabelled set arrives, we simply train the task-specific classifier in a self-supervised manner, while keeping the backbone frozen. For testing we concatenate all the task-specific classifiers, yielding *task-agnostic* inference. The simplicity of our approach lies in the decoupled training on task-specific data, while preserving performance across tasks. We characterize our `Baseline` as *frustratingly simple* as it *neither* requires labelled data, *nor* any specialized losses for preventing forgetting. Additionally, we propose `Baseline++` that stores discovered the novel class prototypes from the previous tasks to further reduce forgetting.

To verify the effectiveness of our proposed baselines, we compare with several state-of-the-art class-iNCD methods [32,40], class-incremental learning methods (CIL) [3,27,30] and unsupervised incremental learning (UIL) [14] methods adapted to the class-iNCD setting. In Fig. 3 we plot the Overall Accuracy ($\mathcal{A}$) and Maximum Forgetting ($\mathcal{F}$) on CIFAR-100 for all the methods under consideration, where higher $\mathcal{A}$ and lower $\mathcal{F}$ is desired from an ideal method. Despite the simplicity, both the `Baseline` and `Baseline++` surprisingly achieve the highest accuracy and least forgetting among all the competitors. Thus, our result sets a precedent to future class-iNCD methods and urge them to meticulously compare with our baselines, that are as simple as having a frozen backbone and a linear classifier.

In a nutshell, our **contributions** are three-fold: (**i**) We bring a paradigm shift in NCD by proposing to use self-supervised pre-trained models as a new starting point, which can substitute the large annotated datasets. (**ii**) We, for the first time, highlight the paramount importance of having strong baselines in class-iNCD, by showcasing that simple baselines if properly implemented can outperform many state-of-the-art methods. To that end, we introduce two baselines (`Baseline` and `Baseline++`) that are simple yet strong. (**iii**) We run extensive experiments on multiple benchmarks and for longer incremental settings.

To foster future research, we release a modular and easily expandable PyTorch repository for the class-iNCD task, that will allow practitioners to replicate the results of this work, as well as build on top of our strong baselines.

2 Related Work

Novel Class Discovery (NCD) was formalized by [18] with the aim of alleviating the innate ambiguity in deep clustering [7,11,48–50] and enhancing the clustering ability of novel classes in an unlabelled dataset, by leveraging the prior knowledge derived from related labelled samples [18,22,23]. Many of the recent NCD works utilize a joint training scheme that assumes access to both labelled and unlabelled data concurrently to exploit strong learning signal from the labelled classes [13,15,19,25,43,51–54].

Keeping in mind the data regulatory practices, the NCD community has been paying more attention to the problem of Incremental Novel Class Discovery (iNCD) [32] where the access to the labelled (or base) dataset is absent during the discovery stage. Unlike iNCD, FRoST [40] and NCDwF [26] investigate a more realistic yet challenging setting known as Class-incremental Novel Class Discovery (class-iNCD), where task-id information is not available during inference. However, all the class-iNCD methods so far have investigated learning in short incremental scenarios (2 steps in [40] and 1 step in [26]). Differently, we explore a more realistic setting of longer incremental setting (up to 5 steps) and show that many existing class-iNCD methods deteriorate in such settings.

Importantly, staying aligned with the original motivation of the NCD and GCD paradigm – *discovering new classes by leveraging prior knowledge* – we propose a new direction to tackle the class-iNCD problem, i.e., by solely leveraging the prior knowledge learned from self-supervised PTMs (e.g., DINO [6]), as opposed to relying on a large amount of expensive highly related *labelled* data.

Class-Incremental Learning (CIL). [35] aims to train a model on a sequence of tasks with access to labelled data only from the current task, while the model's performance is assessed across all tasks it has encountered to date. Notably, the IL methods [3,27,30,38] are devised with a dual objective of mitigating *catastrophic forgetting* [17] of the model's knowledge on the previous tasks, while concurrently enabling it to learn new ones in a flexible manner. To overcome the need of labelled data, unsupervised incremental learning (UIL) [14,31,34] have recently been proposed that aim to learn generalized feature representation via self-supervision to reduce forgetting. Different from UIL, that solely aims to learn a feature encoder, the class-iNCD methods additionally learn a classifier on top of the encoder to classify the unalabelled samples.

Moreover, as shown in the class-iNCD method FRoST [40], due to the differences in the learning objectives of class-iNCD during the supervised pre-training and unsupervised novel class discovery stages, learning continuously is more challenging than the supervised CIL setting. Our proposed baselines attempt to mitigate this issue with cosine normalization of the classifier weights, frozen backbone and feature replay using prototypes, thus greatly simplifying class-iNCD.

3 Method

Problem Formulation. As illustrated in Fig. 1c, a class-iNCD model is trained continuously over T sequential NCD tasks, each of which, $\mathcal{T}^{[\mathtt{t}]}$, presents an unlabelled data set $\mathcal{D}^{[\mathtt{t}]} = \{\boldsymbol{x}_n^{[\mathtt{t}]}\}_{n=1}^{N^{[\mathtt{t}]}}$ with $N^{[\mathtt{t}]}$ instances containing $C^{[\mathtt{t}]}$ novel classes that correspond to a label set $\mathcal{Y}^{[\mathtt{t}]}$. As in prior works [42], we assume that novel classes in $\mathcal{D}^{[\mathtt{i}]}$ and $\mathcal{D}^{[\mathtt{j}]}$ are disjoint, i.e., $\mathcal{Y}^{[\mathtt{i}]} \cap \mathcal{Y}^{[\mathtt{j}]} = \emptyset$. Following the NCD literature, we assume the number of novel classes $C^{[\mathtt{t}]}$ at each step is known as *a priori*. During each discovery step t, we only have access to $\mathcal{D}^{[\mathtt{t}]}$. The aim of class-iNCD is to discover semantically meaningful categories in $\mathcal{D}^{[\mathtt{t}]}$ and accurately group the instances into the discovered clusters, without compromising its performance on the instances from $\mathcal{D}^{[1]}$ to $\mathcal{D}^{[\mathtt{t-1}]}$. In other words, a class-iNCD model comprises a unified mapping function $f\colon \mathcal{X} \rightarrow \bigcup_{t=1}^{T} \mathcal{Y}^{[\mathtt{t}]}$ that can group any test image $\boldsymbol{x}$ into the categories $\bigcup_{t=1}^{T} \mathcal{Y}^{[\mathtt{t}]}$ discovered from the unlabelled task sequence $\boldsymbol{\mathcal{T}} = \{\mathcal{T}^{[1]}, \mathcal{T}^{[2]}, \cdots, \mathcal{T}^{[\mathtt{T}]}\}$ without the help of task-id (i.e., task agnostic inference).

3.1 Overall Framework

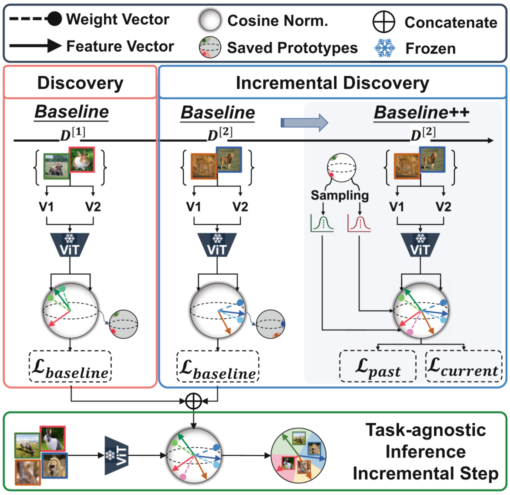


Fig. 4. Overview framework of the proposed methods `Baseline` and `Baseline++` for class-iNCD task.

In this work our goal is to address class-iNCD by leveraging the priors learned by a self-supervised pre-trained model (PTM). To this end we propose a strong baseline called `Baseline` that internally uses the PTM. As illustrated in Fig. 4, the proposed `Baseline` is marked by two steps – (**i**) an initial **discovery step** (see pink box), where task-specific classifier is learned to discover the novel classes contained in $\mathcal{D}^{[1]}$ for the first task $\mathcal{T}^{[1]}$ with a clustering objective ($\mathcal{L}_{\text{baseline}}$). Pseudo per-class prototypes are computed and stored; and (**ii**) it is followed by an **incremental discovery step** (see blue box), where `Baseline` conducts the same discovery training, after which **task-agnostic inference** (see green box) is performed by simply concatenating the two learned task-specific classifiers. `Baseline++` further fine-tunes the concatenated classifier with $\mathcal{L}_{\text{past}}$ and $\mathcal{L}_{\text{current}}$ using the stored class prototypes to strength class-discrimination among tasks. In the following sections, we first present a comprehensive overview of `Baseline`. Additionally, we introduce an advanced variant of `Baseline`, named `Baseline++`, which incorporates feature replay to further mitigate the issue of forgetting.

Discovery Step. In the introductory discovery task $\mathcal{T}^{[1]}$ (see Fig. 4), we learn a mapping function $f^{[1]}: \mathcal{X}^{[1]} \rightarrow \mathcal{Y}^{[1]}$ in a self-supervised manner (i.e., using the Sinkhorn-Knopp cross-view pseudo-labelling [5]) to discover the $C^{[1]}$ categories contained in the given unlabelled data set $\mathcal{D}^{[1]}$. The mapping function $f^{[1]} = h^{[1]} \circ g$ is modeled by a *frozen* feature extractor $g(\cdot)$ and a *Cosine Normalized* linear layer $h^{[1]}(\cdot)$ as task-specific classifier. The $g(\cdot)$ is initialized by the PTM weights θ_g [6], while $h^{[1]}(\cdot)$ is randomly initialized. In other words, only the classifier $h^{[1]}(\cdot)$ weights are learned during this step.

Incremental Discovery Step. After the first discovery step, $\mathcal{D}^{[1]}$ is discarded, and access to only $\mathcal{D}^{[2]}$ is given in the first *incremental* discovery step $\mathcal{T}^{[2]}$ (see Fig. 4). Same as the first step, we train a task-specific mapping function modeled by $f^{[2]} = h^{[2]} \circ g$. The $h^{[2]}$ is newly initialized for the $C^{[2]}$ novel classes of $\mathcal{D}^{[2]}$, while the *frozen* g is shared across tasks. Thanks to the *frozen* feature extractor and *Cosine Normalization* (CosNorm), `Baseline` easily forms a unified model $f^{[1:2]} = h^{[1:2]} \circ g$ by sharing the feature extractor g, and concatenating the two task-specific heads $h^{[1:2]}(\cdot) = h^{[1]}(\cdot) \oplus h^{[2]}(\cdot)$ for task-agnostic inference.

Task-Agnostic Inference. After training for T steps, the inference on the test samples, belonging to any class presented in $\boldsymbol{\mathcal{T}}$, is carried out with the final unified model $f^{[1:\mathrm{T}]} = h^{[1:\mathrm{T}]} \circ g$ in a task-agnostic manner (see Fig. 4).

3.2 Why Use Self-supervised Pre-trained Models?

Table 1. Analysis of NCD accuracy using the same backbone (ViT-B/16) with different pre-training settings.

Pre-training	CIFAR-10	CIFAR-100	CUB-200	Avg. (Δ)
	(5–5)	(50–50)	(100–100)	
Supervised	82.1	32.4	12.8	42.4
PTM-DINO	**95.0**	65.6	36.1	65.6 (+23.2%)
PTM-DINO + Supervised	94.5	**67.2**	**42.5**	**68.1** (+25.7%)

Before delving into the specifics of our method, we first validate the benefits of leveraging self-supervised PTMs for NCD, where supervised pre-training is the standard practice. Specifically, we conduct experiments with our `Baseline` method under traditional NCD setting and splits [15] on three benchmarks (CIFAR-10, CIFAR-100 and CUB-200), comparing three pre-training strategies: (i) supervised pre-training on the labelled set starting from a randomly initialized model (as in Fig. 1a), (ii) self-supervised PTM initialization (e.g., DINO [6], a *self-supervised* model) (as in Fig. 1c), and (iii) supervised fine-tuning starting from PTM initialization. After this step the novel classes are discovered in the unlabelled set. In Table 1 we can see that the PTM-DINO, a model trained without any supervision, performs significantly better in discovering novel classes compared to the supervised counterpart (by +23.2%), which is trained on the highly related base classes. This demonstrates that the original motivation of using a highly related labelled set to aid NCD [18] is clearly suboptimal when compared with self-supervised pre-training on a rather larger dataset. Additionally, fine-tuning PTM-DINO on the labelled samples only gives limited accuracy gain, with the PTM-DINO performing reasonably at-par (-2.5%). Guided by these observations, we propose using strong PTMs (e.g., DINO [6]) with Vision Transformers (ViT) [12] as a new starting point for NCD and class-iNCD, thereby eliminating the dependence on the labelled data.

3.3 Strong Baselines for class-iNCD

In this section we detail the proposed methods, `Baseline` and `Baseline++`, for solving the class-iNCD task. Both the baselines use PTMs, as backbone, that are general purpose and publicly available. Additionally, the `Baseline++` uses latent feature replay. The baselines have been designed to preserve stability on the *past* novel classes, while being flexible enough to discover the novel classes in the *current* task.

Baseline

Self-supervised Training for Discovery. Starting from a frozen feature extractor g, initialized with the weights from DINO [6], we optimize a *self-supervised* clustering objective to directly discover the novel categories at each step. In details, first we randomly initialize a learnable linear layer $h^{[\mathtt{t}]}$ as the task-specific classifier for the $C^{[\mathtt{t}]}$ novel classes contained in the unlabelled set $\mathcal{D}^{[\mathtt{t}]}$. To learn the task-specific network $f^{[\mathtt{t}]}$ for discovery, Baseline employs the Sinkhorn-Knopp cross-view pseudo-labeling algorithm [5]. We optimize a *swapped* prediction problem, where the 'code' $\boldsymbol{y}_1$ of one view is predicted from the representation of another view $\boldsymbol{z}_2$, derived from the same image $\boldsymbol{x}$ through different image transformations, and vice-versa:

$$\mathcal{L}_{\mathtt{Baseline}} = \ell(\boldsymbol{z}_2, \boldsymbol{y}_1) + \ell(\boldsymbol{z}_1, \boldsymbol{y}_2) \tag{1}$$

where $\ell(\cdot,\cdot)$ is the standard cross-entropy loss. We obtain the codes (or *soft-targets*) $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ by using the Sinkhorn-Knopp algorithm. Note that, we freeze the entire feature extractor g during optimizing $\mathcal{L}_{\mathtt{Baseline}}$ as a straightforward way to prevent catastrophic forgetting.

Multi-step Class-Incremental Discovery. Our ultimate goal is to learn a unified mapping function $f^{[\mathtt{1:T}]}\colon \mathcal{X} \rightarrow \bigcup_{t=1}^{T} \mathcal{Y}^{[\mathtt{t}]}$. If all the training data are available, an ideal clustering objective for $f^{[\mathtt{1:T}]}$ can be achieved by minimizing an adequate loss $\mathcal{L}^{[\mathtt{1:T}]}$ at the end of the task sequence:

$$\mathcal{L}^{[\mathtt{1:T}]} = \mathbb{E}_{\mathcal{T}^{[\mathtt{t}]} \sim \boldsymbol{\mathcal{T}}} \mathcal{L}^{[\mathtt{t}]} \tag{2}$$

However, due to the data unavailability of past tasks in class-iNCD, we can only pursue an approximation of this ideal joint objective defined by Eq. 2. In this work, unlike most of the CIL solutions [44], we pursue a better approximation from a new perspective: balancing the individual clustering objectives in each task to a unified importance. To be more specific, the proposed Baseline adopts *frozen feature extractor* with *cosine normalized classifier* to unify the clustering objectives across tasks.

Frozen Feature Extractor. In Baseline we freeze the entire PTM g by introducing $\|\theta_g^{[\mathtt{t}]} - \theta_g^{[\mathtt{t-1}]}\|^2 = 0,\ t \in \{1, \ldots, T\}$ as a constraint. This enables us to leverage the power of the generalist PTM g for all tasks *equally*, without introducing bias towards any particular task, i.e., avoiding the *model drift* issue in CIL literature [46].

Cosine Normalization. The frozen feature extractor not only preserves the powerful prior knowledge from the pre-training data, but also maintains the cooperative mechanism between g and each individual classifier $h^{[\mathtt{t}]}$. With the stable cooperative mechanism, the test data can be directly routed to the corresponding task-specific function network $f = h^{[\mathtt{t}]} \circ g$, if the task-id t is available.

However, the task-id is not available in class-iNCD. To achieve simple task-agnostic inference, we propose to apply *Cosine Normalization* (CosNorm) [21,33] on each individual linear classifier $h^{[\mathtt{t}]}$. This enables the learned classifiers to output scores of the same scale, avoiding imbalance between the past and current novel classes.

Formally, given an input vector $\boldsymbol{x}$, the L2 normalization operation can be defined as $\widetilde{\boldsymbol{x}} = L2Norm(\boldsymbol{x}) = {}^{\boldsymbol{x}}/_{\|\boldsymbol{x}\|} = {}^{\boldsymbol{x}}/_{\sqrt{\boldsymbol{x}\boldsymbol{x}^T+\epsilon}}$, where ϵ is a small value to avoid division by zero and is set to $1\mathrm{e}^{-12}$ in this work. At every discovery step, $L2Norm(\cdot)$ is continuously applied to both the input feature embedding $\boldsymbol{z}$ and each weight vector θ_h^i of the task-specific linear classifier $h^{[\mathtt{t}]}$. $\theta_h^i \in \mathbb{R}^k$ is the i-th column of the classifier weight matrix θ_h, corresponding to one semantic cluster. Consequently, the i-th output logit from the classifier is then computed as:

$$\boldsymbol{l}^i = \widetilde{\theta}_h^{iT} \widetilde{\boldsymbol{z}} = \frac{\theta_h^{iT} \boldsymbol{z}}{\|\theta_h^i\| \|\boldsymbol{z}\|} = cos(\theta_h^i) \tag{3}$$

where $\|\theta^i_{h^{[\mathtt{t}]}}\| = \|\boldsymbol{z}\| = 1$ and $cos(\theta^i_{h^{[\mathtt{t}]}})$ is the cosine similarity between the feature vector $\boldsymbol{z}$ and the i-th weight vector $\theta^i_{h^{[\mathtt{t}]}}$. We thus use the term *CosNorm* for this operation. The magnitude of the output logits $\boldsymbol{l}$ is thereby unified to the same scale $[-1, 1]$ for all classifiers from different steps.

Task-Agnostic Inference. Having the balanced classifier weights, we can then build a unified classification head $h^{[\mathtt{1:T}]}$ by simply concatenating the task-specific heads learned at each step $h^{[\mathtt{1:T}]} = h^{[\mathtt{1}]} \oplus h^{[\mathtt{2}]} \oplus \ldots \oplus h^{[\mathtt{T}]}$. By means of the frozen feature extractor and feature normalization, all the feature embedding $\widetilde{\boldsymbol{z}}$ are mapped to the uniform feature space under the same scale. Incorporating with the normalized classifier weights in $h^{[\mathtt{1:T}]}$, task-agnostic inference can be fairly achieved using $f^{[\mathtt{1:T}]} = h^{[\mathtt{1:T}]} \circ g$ for all the discovered classes so far.

Baseline++

To take full advantage of the stable feature extractor, we propose `Baseline++` that additionally uses the learned model $f^{[\mathtt{t-1}]} = h^{[\mathtt{t-1}]} \circ g$ to compute the pseudo per-class feature prototypes $\boldsymbol{\mu}_{\hat{c}^{[\mathtt{t-1}]}}$ and variances $\boldsymbol{v}^2_{\hat{c}^{[\mathtt{t-1}]}}$ as *proxies* for the novel classes discovered from the previous task $\mathcal{T}^{[\mathtt{t-1}]}$. For the subsequent tasks, features drawn from the Gaussian distribution, constructed with the stored $\boldsymbol{\mu}_{\hat{c}^{[\mathtt{t-1}]}}$ and $\boldsymbol{v}^2_{\hat{c}^{[\mathtt{t-1}]}}$, are replayed to reduce forgetting in the classifiers. We call this simplified replay mechanism as *Knowledge Transfer with Robust Feature Replay* (KTRFR) (see Fig. 4), which we describe next.

Knowledge Transfer with Robust Feature Replay (KTRFR). At each previous discovery step $t \in \{1, \ldots, T-1\}$, `Baseline++` computes and stores a set $\boldsymbol{M}^{[\mathtt{t}]} = \{\mathcal{N}(\boldsymbol{\mu}_{\hat{c}_j^{[\mathtt{t}]}}, \boldsymbol{v}^2_{\hat{c}_j^{[\mathtt{t}]}})\}_{j=1}^{\mathcal{C}^{[\mathtt{t}]}}$ that contains pseudo per-class feature prototype distributions derived from the unlabelled set $\mathcal{D}^{[\mathtt{t}]}$. Here, $\boldsymbol{\mu}_{\hat{c}_j^{[\mathtt{t}]}}$ and $\boldsymbol{v}^2_{\hat{c}_j^{[\mathtt{t}]}}$ are the calculated mean and variance of the feature embedding predicted by the task-specific model $f^{[\mathtt{t}]}$ as pseudo class $\hat{c}_j^{[\mathtt{t}]}$. Since the feature prototype set $\boldsymbol{M}^{[\mathtt{t}]}$ can represent and simulate the novel classes discovered at each previous step, `Baseline++` can further train the concatenated model $f^{[\mathtt{1:T}]} = h^{[\mathtt{1:T}]} \circ g$ by replaying the per-class features sampled from the saved Gaussian distributions in $\{\boldsymbol{M}^{[1]}, \dots, \boldsymbol{M}^{[\mathtt{T}-1]}\}$ with the objective defined as:

$$\mathcal{L}_{\text{past}} = -\mathbb{E}_{\boldsymbol{M}^{[\mathtt{t}]} \sim \boldsymbol{M}^{[\mathtt{1:T}-1]}} \mathbb{E}_{(\boldsymbol{z}^{\hat{c}^{[\mathtt{t}]}}, \hat{\boldsymbol{y}}^{\hat{c}^{[\mathtt{t}]}}) \sim \mathcal{N}(\boldsymbol{\mu}_{c^{[\mathtt{t}]}}, \boldsymbol{v}^2_{c^{[\mathtt{t}]}})} \sum_{j=1}^{\mathcal{C}^{[\mathtt{t}]}} \hat{\boldsymbol{y}}^{\hat{c}_j^{[\mathtt{t}]}} \log \sigma(\frac{h^{[\mathtt{1:T}]}(\boldsymbol{z}^{\hat{c}_j^{[\mathtt{t}]}})}{\tau}) \quad (4)$$

where, $\sigma(\cdot)$ is a softmax function and τ is the temperature. By optimizing the objective defined in Eq. 4, `Baseline++` can better approximate the ideal objective defined in Eq. 2 by simulating the past data distribution. Furthermore, to maintain the clustering performance for the current novel classes in $\mathcal{D}^{[\mathtt{T}]}$, we also transfer the knowledge from the current task-specific head $h^{[\mathtt{T}]}$ to $h^{[\mathtt{1:T}]}$. In details, using the pseudo-labels $\hat{\boldsymbol{y}}_i^{[\mathtt{T}]}$ computed by the learned $f^{[\mathtt{T}]}$, we can build a pseudo-labelled data set $\mathcal{D}_{PL}^{[\mathtt{T}]} = \{\boldsymbol{x}_i^{[\mathtt{T}]}, \hat{\boldsymbol{y}}_i^{[\mathtt{T}]}\}_{i=1}^{N^{[\mathtt{T}]}}$. The task-specific knowledge stored in the pseudo-labels can be then transferred to the unified classifier by optimizing the following objective:

$$\mathcal{L}_{\text{current}} = -\mathbb{E}_{(\boldsymbol{x}^{[\mathtt{T}]}, \hat{\boldsymbol{y}}^{[\mathtt{T}]}) \sim \mathcal{D}_{PL}^{[\mathtt{T}]}} \sum_{j=1}^{C^{[\mathtt{T}]}} \hat{y}_j^{c^{[\mathtt{T}]}} \log \sigma(\frac{h^{[\mathtt{1:T}]}(g(\boldsymbol{x}^{c^{[\mathtt{T}]}}))}{\tau}). \quad (5)$$

The final *past-current* objective for KTRFR training at step T of `Baseline++` is formulated as:

$$\mathcal{L}_{\mathtt{Baseline++}} = \mathcal{L}_{\text{past}} + \mathcal{L}_{\text{current}} \quad (6)$$

4 Experiments

4.1 Experimental Settings

Datasets and Splits. We conduct experiments on three generic image recognition datasets and two fine-grained recognition datasets: CIFAR-10 (C10) [28], CIFAR-100 (C100) [28], TinyImageNet-200 (T200) [29], CUB-200 (B200) [45] and Herbarium-683 (H683) [41]. Although the PTM (DINO) used in our baselines and the methods we compared was pre-trained without labels, there's a potential for category overlap between the pre-training dataset (ImageNet [10]) and C10, C100, and T200. To ensure a equitable evaluation, we include B200 and H683 datasets. Notably, B200 shares only two categories with DINO's pre-training dataset (ImageNet), whereas H683 has no overlap whatsoever. For each dataset, we adopt two strategies (two-step and five-step) to generate the task sequences, where the total classes and corresponding instances of training data are divided averagely for each step. The test data are used for evaluation. Detailed data splits are provided in the supplementary material.

Evaluation Protocol. We evaluate all the methods in class-iNCD using the **task-agnostic** evaluation protocol [40]. Specifically, we do not know the task

ID of the test sample during inference, and the network must route the sample to the correct segment of the unified classifier.

Evaluation Metrics. We report two metrics: maximum forgetting $\mathcal{F}$ and overall discovery accuracy (or clustering accuracy [40]) $\mathcal{A}$ for all discovered classes by the end of the task sequence. $\mathcal{F}$ measures the difference in clustering accuracy between the task-specific model $f^{[1]}$ and the unified model $f^{[\mathtt{1:T}]}$ (at the last step) for samples belonging to novel classes discovered at the first step. $\mathcal{A}$ is the clustering accuracy from the unified model $f^{[\mathtt{1:T}]}$ on instances from all the novel classes discovered by the end of the sequence.

4.2 Implementation Details

`Baseline` **and** `Baseline++`**.** By default, ViT-B/16 [12] is used as the backbone g with DINO [6] initialization for all data sets. The 768-dimensional output vector $\boldsymbol{z} \in \mathbb{R}^{768}$, from the $[CLS]$ token is used as the deep features extracted from a given image. g is frozen during training. Following the backbone, one *cosine normalized* linear layer (without bias) is randomly initialized as the task-specific classifier $h^{[\mathtt{t}]}$ with $\mathcal{C}^{[\mathtt{t}]}$ output neurons. Soft pseudo-labels self-supervised are generated using the Sinkhorn-Knopp [5,9] algorithm with default hyperparameters (number of iterations = 3 and $\epsilon = 0.05$).

Training. At each step, we train the model for 200 epochs on the given unlabelled data set $\mathcal{D}^{[\mathtt{t}]}$ with the same data augmentation strategy [8] in all the experiments. After the discovery stage, `Baseline++` further conducts KTRFR training on the unified model $f^{[\mathtt{1:t}]}$ for 200 epochs. A cosine annealing learning rate scheduler with a base rate of 0.1 is used. The model is trained on mini-batches of size 256 using SGD optimizer with a momentum of 0.9 and weight decay 10^{-4}. The temperature τ is set to 0.1.

4.3 Analysis and Ablation Study

Comparison with Reference Methods. We first establish reference methods using K-means [1] and joint training scheme (`Joint (frozen)`, based on `Baseline` but access to the previous training data is given) [30], respectively.

To further enhance the upper reference performance, we unfreeze the last transformer block during training on joint data sets, which is referred as to `Joint (unfrozen)` method. As shown in Table 2, the joint training methods slightly outperform our baselines on all data sets and splits, as they can jointly optimize the ideal objective defined in Eq. 2 using the given access to all training data. Nonetheless, our baselines perform nearly as well as the joint training methods, indicating limited benefits from access to all unlabelled data under class-iNCD and the effectiveness of our baselines.

Table 2. Comparison of our proposed baselines with reference methods on two task splits of C10, C100 and T200.

	Datasets	C10		C100		T200	
	Methods	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$
Two-step	Kmeans [24]	93.9	87.3	68.2	56.7	62.0	47.1
	Joint (frozen)	4.9	92.1	5.3	61.8	3.3	51.1
	Joint (unfrozen)	**0.8**	**92.4**	**2.5**	**65.2**	2.3	**56.5**
	`Baseline`	8.5	89.2	6.7	60.3	4.0	54.6
	`Baseline++`	4.5	90.9	6.6	61.4	**0.2**	55.1
Five-step	Kmeans [24]	99.1	82.1	76.3	54.3	66.0	52.9
	Joint (frozen)	5.1	93.8	10.5	68.6	1.8	57.8
	Joint (unfrozen)	**1.5**	**97.5**	**5.9**	**74.9**	3.0	**60.7**
	`Baseline`	8.2	85.4	15.6	63.7	9.2	53.3
	`Baseline++`	7.6	91.7	12.3	67.7	**1.6**	56.5

Table 3. Self-ablation analysis of the proposed components on two task splits of C10, C100 and T200.

		Datasets		C10		C100		T200	
		CosNorm	KTRFR	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$
Two-step	(a)	✓	✓	**4.5**	**90.9**	6.6	**61.4**	**0.2**	**55.1**
	(b)	✓	✗	8.5	89.2	6.7	60.3	4.0	54.6
	(c)	✗	✓	8.2	80.2	**5.1**	54.1	3.3	38.9
	(d)	✗	✗	16.1	74.3	7.3	50.1	4.3	33.2
Five-step	(a)	✓	✓	7.6	**91.7**	**12.3**	**67.7**	1.6	**56.5**
	(b)	✓	✗	8.2	85.4	15.6	63.7	9.2	53.3
	(c)	✗	✓	**6.3**	90.7	14.3	58.2	**0.7**	49.7
	(d)	✗	✗	10.9	80.3	16.6	49.1	8.1	41.9

Ablation on Proposed Components. We further present an ablation study on the individual core components of our baseliens, namely CosNorm and KTRFR. Results are shown in Table 3. It is noticeable from the results that CosNorm plays a substantial role in enhancing the overall accuracy of our proposed baselines (refer to `Baseline`: *(b) v.s. (d)* and `Baseline++`: *(a) v.s. (c)*). This is attributed to its unification capability to effectively address the issue of that the weight vectors with significant magnitudes in $f^{[1:T]} = h^{[1:T]} \circ g$ always dominating the prediction. On the other hand, KTRFR can improve the overall accuracy and mitigate the forgetting at the end of each task sequence (refer to *(a) v.s. (b)* and *(c) v.s. (d)*). Of particular note is that the performance gain attained by using KTRFR is more significant when dealing with longer task sequences (refer to the *upper half v.s. lower half* in Table 3). `Baseline++` (a) equipped with both CosNorm and KTRFR achieves the best overall accuracy and the least forgetting.

Analysis of Pre-trained Models (PTM). In Table 4, we present a comparison between different PTMs such as ResNet50 [20] and ViT-B/16 [12], along with various pre-training strategies (CLIP [37] and DINO [6]). Transformer architecture achieves superior performance owing to its discrimination ability [36]. CLIP pre-training achieves similar outcomes to DINO, demonstrating the effectiveness of strong PTM with a different pre-training strategy on web data.

Table 4. Ablation of architectures and pre-training strategies of PTMs on five-step splits of C10, C100 and T200.

	`Baseline`					
Datasets	C10		C100		T200	
Backbones	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$
ResNet50-DINO	37.5	45.8	16.4	38.5	10.1	24.7
ViT-B/16-DINO	8.2	85.4	**15.6**	**63.7**	**9.2**	**53.3**
ViT-B/16-CLIP	**5.3**	**87.5**	17.1	62.4	15.7	42.5

4.4 Comparison with the State-of-the-Art Methods

For a comprehensive comparison, we adapt methods from closely related fields for state-of-the-art comparison. We adjust ResTune [32] and FRoST [40] to the multi-step class-iNCD setting from the closely related iNCD field. We adapt three representative CIL methods: EwC [27], LwF [30], and DER [3] to this self-supervised setting. Similarly, we adapt the UIL method, CaSSLe [14], for incremental discovery. All adapted methods employ ViT-B/16 with the same DINO-initialization as a feature extractor. For the adapted CIL and UIL methods, the same self-training strategy is used as in our `Baseline` method to prevent forgetting. All adapted methods unfreeze only the last transformer block of the feature extractor [2,47], except ResTune that unfreezes the last two blocks for model growing. More implementation details can be found in the supplementary material.

Table 5 compares our proposed `Baseline` and `Baseline++` with the adapted methods. ResTune underperforms in the class-iNCD setting due to its reliance on task-id information. FRoST exhibits strong ability to prevent forgetting on all data sets and sequences by segregating the *not-forgetting* objective between the feature extractor and classifier. The adapted CIL methods capably discover new classes leveraging PTM knowledge. For two-step split sequences, these methods generally outperform class-iNCD adaptations by maintaining a balance between old and new classes. However, on five-step split sequences, the advantage of CIL-based methods over class-iNCD-based methods is not evident anymore, because CIL-based methods tend to forget tasks at the initial steps more when dealing with long sequences, as widely studied in CIL literature. EwC achieves better discovery accuracy by applying its forgetting prevention component directly

Table 5. Comparison with the adapted state-of-the-art methods on two task splits of C10, C100, T200, B200, and H683 under class-iNCD setting with the same DINO-ViT-B/16 backbone. Overall accuracy and maximum forgetting are reported.

	Datasets	C10		C100		T200		B200		H683	
	Methods	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$	$\mathcal{F}\downarrow$	$\mathcal{A}\uparrow$
Two-step	EwC [27]	32.4	79.0	42.5	43.9	27.2	33.3	18.1	25.5	13.8	25.1
	LwF [30]	30.4	34.4	44.1	42.4	40.0	27.2	20.2	23.9	16.3	24.9
	DER [3]	49.0	69.9	29.8	30.3	39.0	28.9	5.0	20.4	14.0	24.7
	ResTune [32]	97.6	47.2	32.7	17.1	32.3	17.2	12.0	13.0	27.4	17.1
	FRoST [40]	**2.5**	46.6	**4.7**	34.2	4.3	26.1	**3.9**	17.6	16.2	18.4
	CaSSLe [14]	9.1	87.3	10.3	53.7	6.9	36.5	4.8	26.8	10.9	25.3
	`Baseline`	8.5	**89.2**	6.7	**60.3**	4.0	**54.6**	4.1	**28.7**	7.9	**25.7**
	`Baseline++`	4.5	**90.9**	6.6	**61.4**	**0.2**	**55.1**	4.2	**36.9**	**6.0**	**27.5**
Five-step	EwC [27]	21.1	81.1	60.1	30.6	48.0	23.2	21.2	19.1	15.7	22.4
	LwF [30]	20.1	25.8	60.9	16.1	53.7	15.6	21.7	15.7	16.5	23.4
	DER [3]	30.1	76.2	62.6	36.2	52.1	21.7	16.2	16.3	18.0	22.3
	ResTune [32]	95.5	49.2	83.3	19.4	60.4	12.2	24.2	12.4	28.2	11.2
	FRoST [40]	**0.9**	69.2	14.2	43.6	14.4	31.0	19.4	18.5	13.5	23.4
	CaSSLe [14]	11.3	78.5	25.3	61.7	14.1	42.3	14.6	22.3	13.8	24.1
	`Baseline`	8.2	**85.4**	15.6	**63.7**	9.2	**53.3**	13.7	**28.9**	3.1	**25.2**
	`Baseline++`	7.6	**91.7**	**12.3**	**67.7**	**1.6**	**56.5**	**0.6**	**41.1**	**2.7**	**26.1**

to the model parameters using Fisher information matrix, while LwF [30] faces slow-fast learning interference issues. DER's performance suffers due to unstable self-supervised trajectories. CaSSLe is notably proficient in incremental discovery, attributed to its effective distillation mechanisms. Without *bells* and *whistles*, our `Baseline` and `Baseline++` models consistently outperform adapted methods across datasets and sequences. While FRoST gives lower forgetting in some two-step split cases, our `Baseline++`, by improving the capacity for class-discrimination across all tasks, achieves lower forgetting in most five-step split cases.

Generalizability Analysis. Our proposed approach offers a versatile framework to convert related methods into effective class-iNCD solutions. In Fig. 5, we equip two such methods, AutoNovel [19] and OCRA [4], with our proposed components (frozen PTM and CosNorm). The results emphasize the pivotal role of CosNorm in forming a task-agnostic classifier. Our findings reveal that, by removing CosNorm, the converted methods suffer from significant forgetting due to non-uniformly scaled weight vectors, resulting in a decrease in overall discovery accuracy. This echoes the importance of CosNorm in aligning the magnitude of the classifiers learned at each step to the same scale in class-iNCD scenarios. Instead, with using CosNorm, PTMs can be effectively leveraged to develop strong methods for the problem of class-iNCD.

5 Conclusion

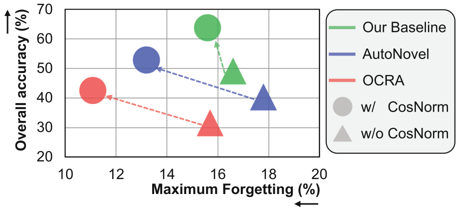


Fig. 5. Generalizability analysis. Results are reported on the five-step split of C100 with DINO-ViT-B/16.

In this work we address the practical yet challenging task of Class-incremental Novel Class Discovery (class-iNCD). First, we highlight that the use of self-supervised pre-trained models (PTMs) can achieve better or comparable performance to models trained with labelled data in NCD. Building upon this observation, we propose to forego the need for expensive labelled data by leveraging PTMs for class-iNCD. Second, we introduce two simple yet strong baselines that comprise of frozen PTM, cosine normalization and knowledge transfer with robust feature replay. Notably, our proposed baselines demonstrate significant improvements over the state-of-the-art methods across five datasets. We hope our work can provide a new, promising avenue towards effective class-iNCD.

Acknowledgements. E.R. is supported by MUR PNRR project FAIR - Future AI Research (PE00000013), funded by NextGenerationEU and EU projects SPRING (No. 871245) and ELIAS (No. 01120237). M.L. is supported by the PRIN project LEGO-AI (Prot. 2020TA3K9N). This work was carried out in the Vision and Learning joint laboratory of FBK and UNITN.

References

1. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: ACM-SIAM Symposium on Discrete Algorithms (2007)
2. Boschini, M., et al.: Transfer without forgetting. In: ECCV (2022)
3. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. In: NeurIPS (2020)
4. Cao, K., Brbic, M., Leskovec, J.: Open-world semi-supervised learning. arXiv (2021)
5. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
6. Caron, M., et al.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
7. Chang, J., Wang, L., Meng, G., Xiang, S., Pan, C.: Deep adaptive image clustering. ICCV (2017)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. arXiv (2020)
9. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: NeurIPS (2013)

10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR (2009)
11. Dizaji, K.G., Herandi, A., Deng, C., Cai, W.T., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: ICCV (2017)
12. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv (2020)
13. Fei, Y., Zhao, Z., Yang, S.X., Zhao, B.: Xcon: learning with experts for fine-grained category discovery. In: BMVC (2022)
14. Fini, E., Costa, V., Alameda-Pineda, X., Ricci, E., Karteek, A., Mairal, J.: Self-supervised models are continual learners. In: CVPR (2022)
15. Fini, E., Sangineto, E., Lathuilière, S., Zhong, Z., Nabi, M., Ricci, E.: A unified objective for novel class discovery. In: ICCV (2021)
16. Fomenko, V., Elezi, I., Ramanan, D., Leal-Taixé, L., Osep, A.: Learning to discover and detect objects. In: NeurIPS (2022)
17. French, R.: Catastrophic forgetting in connectionist networks. Trends Cogn. Sci. (1999)
18. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: ICCV (2019)
19. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. In: ICLR (2020)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2015)
21. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR (2019)
22. Hsu, Y.C., Lv, Z., Kira, Z.: Learning to cluster in order to transfer across domains and tasks. arXiv (2017)
23. Hsu, Y.C., Lv, Z., Schlosser, J., Odom, P., Kira, Z.: Multi-class classification without multi-class labels. arXiv (2019)
24. Jain, A.K.: Data clustering: 50 years beyond k-means. In: PRL (2008)
25. Jia, X., Han, K., Zhu, Y., Green, B.: Joint representation learning and novel category discovery on single- and multi-modal data. In: ICCV (2021)
26. Joseph, K.J., et al.: Novel class discovery without forgetting. In: ECCV (2022)
27. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. In: Proceedings of the National Academy of Sciences (2016)
28. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
29. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
30. Li, Z., Hoiem, D.: Learning without forgetting. In: TPAMI (2017)
31. Lin, Z., Wang, Y., Lin, H.: Continual contrastive learning for image classification. In: 2022 IEEE International Conference on Multimedia and Expo (ICME) (2022)
32. Liu, Y., Tuytelaars, T.: Residual tuning: toward novel category discovery without labels. In: TNNLS (2022)
33. Luo, C., Zhan, J., Wang, L., Yang, Q.: Cosine normalization: using cosine similarity instead of dot product in neural networks. arXiv (2017)
34. Madaan, D., Yoon, J., Li, Y., Liu, Y., Hwang, S.J.: Representational continuity for unsupervised continual learning. In: ICLR (2022). https://openreview.net/forum?id=9Hrka5PA7LW

35. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., Van De Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
36. Naseer, M., Ranasinghe, K., Khan, S.H., Hayat, M., Khan, F.S., Yang, M.H.: Intriguing properties of vision transformers. In: NeurIPS (2021)
37. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
38. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCarl: incremental classifier and representation learning. In: CVPR (2016)
39. Riz, L., Saltori, C., Ricci, E., Poiesi, F.: Novel class discovery for 3D point cloud semantic segmentation. In: CVPR (2023)
40. Roy, S., Liu, M., Zhong, Z., Sebe, N., Ricci, E.: Class-incremental novel class discovery. arXiv (2022)
41. Tan, K.C., Liu, Y., Ambrose, B.A., Tulig, M.C., Belongie, S.J.: The herbarium challenge 2019 dataset. arXiv (2019)
42. Troisemaine, C., Lemaire, V., Gosselin, S., Reiffers-Masson, A., Flocon-Cholet, J., Vaton, S.: Novel class discovery: an introduction and key concepts. arXiv (2023)
43. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: CVPR (2022)
44. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: theory, method and application. arXiv (2023)
45. Welinder, P., et al.: Caltech-UCSD birds 200 (2010)
46. Wu, T.Y., et al.: Class-incremental learning with strong pre-trained models. In: CVPR (2022)
47. Wu, Y., et al.: Large scale incremental learning. In: CVPR (2019)
48. Xie, J., Girshick, R.B., Farhadi, A.: Unsupervised deep embedding for clustering analysis. arXiv (2015)
49. Yang, B., Fu, X., Sidiropoulos, N., Hong, M.: Towards k-means-friendly spaces: simultaneous deep learning and clustering. In: ICML (2016)
50. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: CVPR (2016)
51. Yang, M., Zhu, Y., Yu, J., Wu, A., Deng, C.: Divide and conquer: compositional experts for generalized novel class discovery. In: CVPR (2022)
52. Zhao, B., Han, K.: Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. arXiv (2021)
53. Zhong, Z., Fini, E., Roy, S., Luo, Z., Ricci, E., Sebe, N.: Neighborhood contrastive learning for novel class discovery. In: CVPR (2021)
54. Zhong, Z., Zhu, L., Luo, Z., Li, S., Yang, Y., Sebe, N.: Openmix: reviving known knowledge for discovering novel visual categories in an open world. In: CVPR (2020)
55. Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., Calderara, S.: Class-incremental continual learning into the extended der-verse. In: TPAMI (2022)
56. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. In: IJCV (2020)
57. Grill, J.B., et al.: Bootstrap your own latent-a new approach to self-supervised learning. In: NeurIPS (2020)

Layer-Wise Pruning Ratios Auto-configuration: A One-Shot Channel Pruning Through Sensitivity and Spatial Analysis

Guonan Li[1], Meibao Yao[1](✉), and Xueming Xiao[2]

[1] Jilin University, Changchun 130012, China
meibaoyao@gmail.com
[2] Changchun University of Science and Technology, Changchun 130022, China

Abstract. Pruning is a common compression approach for neural networks. Existing structured pruning methods suffer two major issues: (1) coupling of analyzing pruning ratios configuration and which specific convolution kernels to remove from a certain layer, i.e., intra-layer pruning strategy, makes it difficult to evaluate their real impact, (2) insufficient consideration of spatial context results in the loss of target features, leading to a significant accuracy decline. To this end, we propose a Layer-wise Pruning Ratios Auto-configuration (LPRA) framework, offering a standardized pruning ratios configuration approach for various intra-layer pruning strategies, making it easy to compare their actual pruning effects. We simultaneously assess two metrics: Spatial Coherence (SC), representing the extraction capability of spatial features, and Hessian Mean (HM), representing the sensitivity of convolutional kernels, indicating the impact of the layer on the output. LPRA can preserve spatial information in the model while reduce its impact on accuracy, achieving efficient one-shot channel pruning without introducing any additional network modules or regularization losses. Experimental results show that LPRA performs well in both classification and segmentation tasks, pruning almost 50% of parameters from VGG-16 with 0.36% accuracy gain on CIFAR-100, pruning 41% of parameters from ResNet-34 with 0.16% accuracy gain on CIFAR-10. Furthermore, we identify redundancy in real-time semantic segmentation models.

Keywords: Computer Vision · Convolutional Neural Networks · Model Compression

1 Introduction

Convolutional Neural Networks (CNNs) have demonstrated outstanding performance in the field of computer vision. However, the large number of parameters

This work was sponsored by the National Natural Science Foundation of China (NSFC) through grants No. 62103163 and No. U2341214, and Natural Science Foundation of Shanghai through grant No. 22ZR1479600. We thank the above mentioned funds for their financial support.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 143–159, 2025.
https://doi.org/10.1007/978-3-031-78444-6_10

and computational demands make the deployment of these models in resource-constrained devices and scenarios challenging. CNNs often have excessive parameters, requiring the use of model compression techniques to eliminate redundancies.

Structured pruning, recognized for its simplicity and flexibility, is a widely adopted technique in model compression. It eliminates redundancy from pre-trained neural networks by assessing the significance of convolutional kernels. The number of kernels suitable for removal varies for each convolutional layer. Layers with higher sparsity contain less information, allowing for the pruning of more kernels within the layer. The pruning strategy for convolutional kernels includes direct kernel analysis [12,17,29] and data analysis [22] using gradients or feature maps. Despite post-pruning fine-tuning, traditional methods struggle to significantly surpass the original network's accuracy. Advanced pruning introduces modules or regularization terms in training, promoting sparsity but adding computational overhead and lacking transferability to hardware deployment.

Although structured pruning research is comprehensive, two key issues persist: (1) Traditional methods place excessive emphasis on evaluating the importance of convolutional kernels and overlook the pruning ratios configuration for each layer. We cannot confirm whether the former or the latter improves the pruning effect. Therefore, a unified and standardized pruning ratios configuration scheme is needed to perform pruning for each layer based on this. (2) Previous pruning methods have neglected spatial contextual information, causing the pruned network to lose some learned object features, resulting in significant accuracy reduction, especially in computationally intensive tasks such as semantic segmentation.

Inspired by "Random Channel Pruning" [18], which emphasizes the importance of the pruning configuration, it highlights that even the simplest pruning strategy based on the L1 norm can achieve comparable results to more complex methods when using the same configuration (e.g., pruning ratios, fine-tuning periods, etc.). We view the model as a container, where its capacity determines its learning capacity. Setting the pruning ratios for each layer influences the shape of the container, which is the key factor that affects its capacity. To obtain a reasonable pruning ratios configuration scheme to ensure model capacity, we propose a framework called "Layer Pruning Ratios Auto-configuration" (LPRA). Through a comprehensive analysis of the entire pre-pruned network, LPRA determines the pruning ratios for each layer, providing a standardized pruning ratios configuration method for various intra-layer pruning strategies. LPRA consists of sensitivity analysis and feature extraction capability analysis. "Sensitivity" represents the influence of a certain layer on the final result, while the "feature extraction capability" reflects the ability of layers to capture spatial context information. Figure 1 shows the general framework of our LPRA method. Sensitivity is represented by the Hessian Mean (HM) metric obtained through Hessian analysis, while spatial feature extraction capabilities are quantified using the novel Spatial Coherence (SC) metric. By uncovering the trade-off between HM and SC on the impact of pruning results, we quantified the sparsity of each layer and obtained specific pruning configuration schemes. LPRA gets pruning rate for each layer by one-shot analysis of the network, requiring few annotated data and computational overhead. There's no need to introduce additional modules or loss terms. It works for compressing all networks with convolutional architecture and seamlessly integrates with other model compression techniques.

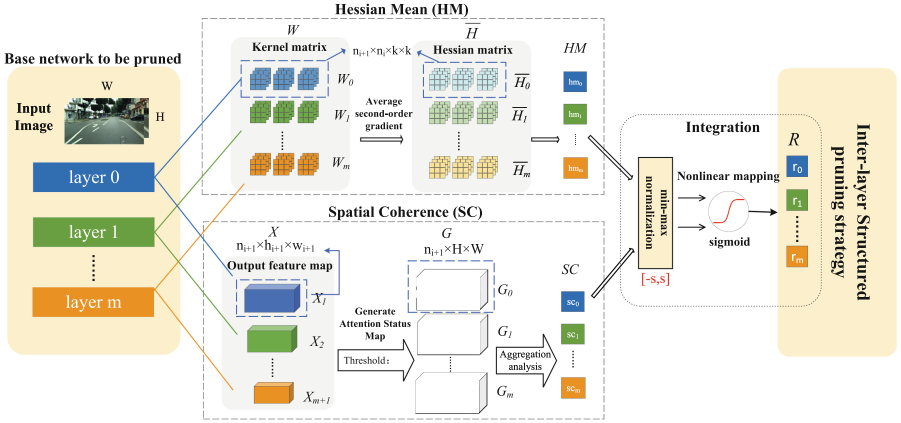


Fig. 1. Our LPRA framework is a standardized pruning ratios configuration approach by analyzing the sensitivity of each layer's convolutional kernel W and the aggregation of output feature maps X, yielding vectors HM and SC. The trade-off between HM and SC on the impact of pruning results is expressed through the module "Integration", resulting in the generation of the pruning rate vector R, which guides layer-specific pruning strategies

We summarize our main contributions as follows:

- We proposed the LPRA framework, which efficiently quantifies the sparsity relationship between convolutional layers. It serves as a standardized and automated pruning ratios configuration method, facilitating the comparison of intra-layer pruning strategies.
- We first introduced the Spatial Coherence (SC) metric to preserve spatial information. Furthermore, we simplified the analysis process by calculating the average Hessian Mean (HM) of the convolutional kernel to depict the sensitivity of the convolutional layer.
- Compare to traditional methods, LPRA demonstrated superior performance in image classification and semantic segmentation tasks. We achieved a 0.36% accuracy improvement while pruning 50% of the parameters with VGG-16 on CIFAR-100.

2 Related Work

Common neural network compression techniques include quantization [14,26], knowledge distillation [7,23], low-rank approximation [1], unstructured pruning [8], and structured pruning. Notably, structured pruning is the only method that preserves the overall architecture of the network. Its hardware-friendly and flexibility in selecting redundant network structures make it a versatile tool for compression, particularly suitable for resource-constrained devices.

Structured pruning typically adheres to the Train-Prune-Finetune paradigm. Initially, the base model is trained, followed by pruning operations on a pre-trained model. To regain accuracy, a specified number of fine-tuning rounds are performed. The pivotal step is the pruning operation, designed to minimize accuracy loss while preserving substantial model capacity.

Within the Train-Prune-Finetune paradigm, pruning can be classified into two types: 1) data-independent and 2) data-dependent. The first approach designs strategies solely based on the analysis of pre-trained model weights, such as L1/L2-based channel pruning [17] and geometric median FPGM [12]. This approach is advantageous for its simplicity, requiring no additional inference or training. The second approach relies on data, analyzing gradients or activation values during training provides a thorough understanding of importance, improving information capture. Evaluation criteria include means of first or second-order Taylor series approximations [9,16,21], feature map sparsity level APoZ [13], and channel impact on output feature maps ThiNet [20]. Traditional methods often focus on individual kernels, lacking a unified benchmark for layer-wise pruning ratios. Recent research [18] highlights issues with pruning method benchmarks. Results suggest that common methods perform similarly under the same random pruning ratios, even L1/L2-based channel pruning comparable to other standards. This shifts our focus towards more holistic pruning ratios configuration strategies.

Recent methods induce neural network sparsity through additional modules [3,11]. However, this brings extra computational challenges and a black-box-like operation for configuring pruning ratios. Previous work [22] proposed a global importance criterion using filter gradients and norms. Other approaches [19,27] introduced sparsity loss beyond task loss, pruning filters below a threshold. Yet, the impact of the regularization term on task loss can bias training. Assessing all kernels with the same criterion is impractical due to significant variations across layers and may lead to irrecoverable accuracy loss. Therefore, an interpretable and unified benchmark is necessary to automate layer-wise pruning ratios.

3 Methodology

3.1 Problem Formulation

Our LPRA method is based on the analysis of convolutional kernels and feature maps at each layer of the convolutional neural network. Here, we provide an explanation of the symbolic definitions. Let $layer_i$ represent the i-th convolutional layer ($0 \leq i \leq m$), where there are m layers to be pruned. n_i represents the number of channels in the input feature map of the i-th layer. Taking a convolutional layer as the unit of interest for measuring layer-wise sparsity, let X_i denote the input feature map of the i-th layer, $X_i \in \mathbb{R}^{n_i \times h_i \times w_i}$, with h_i and w_i as height and width. The corresponding output feature map for the i-th layer is $X_{i+1} \in \mathbb{R}^{n_{i+1} \times h_{i+1} \times w_{i+1}}$, which is also the input for the $(i+1)$-th layer. W_i represents the convolutional kernels for the i-th layer, $W_i \in \mathbb{R}^{n_{i+1} \times n_i \times k_i \times k_i}$, where k is the size of the convolutional kernel, and the kernel corresponding to the j-th output channel is $W_{i,j} \in \mathbb{R}^{n_i \times k_i \times k_i}$. For comprehensive layer-wise sparsity measurement, the LPRA method simultaneously extracts all convolutional kernels W_i and the output feature map X_{i+1}. Two metrics, Hessian Mean (HM) and Spatial Coherence (SC), are calculated for interpretability. Through nonlinear mapping, specific pruning ratios $R = [r_1, r_2, \ldots, r_m]$, where $r_i \in [0, 1)$, are quantified for each layer, yielding an m-dimensional vector.

3.2 Layer-Wise Evaluation Metrics for LPRA

In this section, we will provide a detailed explanation of the design principles, specific calculation methods, and rationale behind the HM and SC metrics.

Hessian Mean (HM). We seek to measure the layer-wise sensitivity of the network, assessing how variations in each layer impact the final network accuracy. Due to the vast parameter volume of convolutional neural networks, an exhaustive algorithm to traverse each layer's influence on the final performance is computationally prohibitive. The L1 norm [17] uses the most basic sensitivity analysis, individual layer sensitivity was evaluated by autonomously pruning each layer and assessing the pruned network's accuracy on the validation set. However, this method requires a significant amount of additional time and computational resources, cannot be considered as a standardized pruning ratios configuration approach. A definitive benchmark is necessary for determining optimal pruning ratios for each layer. The challenges in configuring single-layer pruning ratio resemble those faced in selecting quantization precision for mixed-precision quantization [4,5]. HAWQ [5] proposed an automated method for determining relative quantization precision, using the Hessian spectrum for each layer to establish quantization precision at each hierarchical level. Inspired by HAWQ, utilizing second-order information from the Hessian matrix, rather than first-order information from the gradient vector, provides a more discerning sensitivity measure.

We start by calculating the first-order derivatives of the final task loss L with respect to all parameters in the i-th layer. This results in the gradient matrix g_i for each layer in the network:

$$g_i = \frac{\partial L}{\partial W_i} \tag{1}$$

Next, we calculate second-order derivatives, incorporating a random vector to enhance the generalizability of the Hessian matrix in this process:

$$\frac{\partial(g_i^T v)}{\partial W_i} = \frac{\partial g_i^T}{\partial W_i} v + g_i^T \frac{\partial v}{\partial W_i} = \frac{\partial g_i^T}{\partial W_i} v = H_i v \tag{2}$$

It's important to note that the vector v has the same dimensionality as g_i, and H_i denotes the Hessian matrix of L with respect to W_i.

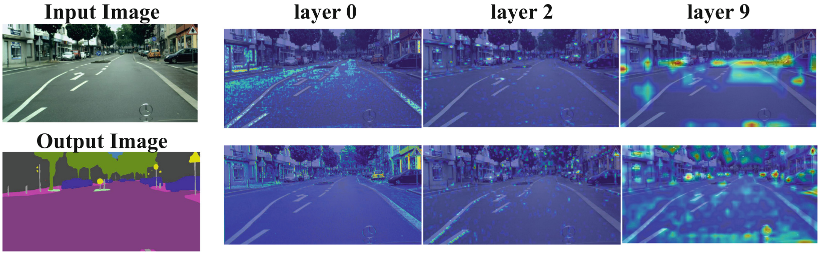


Fig. 2. On Cityscapes dataset, the Class Activation Maps (CAM) for layer0, layer2 and layer9 of the STDC network

This algorithm involves iterative assessments of the Hessian matrix. Distinguishing itself from HAWQ [5], which exclusively retains the top eigenvalues of H_i, our approach entails averaging the results over multiple evaluations, denoted as $\overline{H_i}$.

$$\overline{H_i} = \frac{1}{n} \sum_{i=1}^{n} H_i \tag{3}$$

where n represents the total count of evaluations. Our objective is to evaluate the sensitivity of each layer within the network. Given the substantial variability in the number of channels across layers, a standardized and impartial metric is essential. To address this issue, we obtain the output channel count, represented by n_{i+1}, for $layer_i$ and measure the sensitivity of $layer_i$ by averaging the metric across all channels:

$$hm_i = \frac{\sum_{i=1}^{n_{i+1}} \overline{H_i}}{n_{i+1}} \tag{4}$$

Let hm_i denote the sensitivity of layer i. The computation is carried out for each of the m convolutional layers earmarked for pruning. Consequently, an m-dimensional vector, $HM = [hm_1, hm_2, ..., hm_m]^T$, is obtained. This vector serves as the representation of the layer-wise sensitivities. Larger sensitivity values signify that alterations in each convolutional kernel of the respective layer exert a more substantial influence on the final model output.

Spatial Coherence (SC). We introduce, the Spatial Coherence (SC) metric for the first time to gauge the network's spatial feature extraction capability. Drawing inspiration from Class Activation Mapping (CAM) [28], a technique that delineates regions of interest in images, we visualize attention regions of distinct layers in the real-time semantic segmentation network STDC [6], depicted in Fig. 2. CAM maps exhibit varying degrees of concentration in attention regions, with some being more focused while others are more dispersed. According to our analysis, more concentrated feature representations are deemed to be more targeted, which can be computed using fewer channels, as an excess of channels tends to introduce undesirable noise interference. We are considering introducing a metric to measure the model's aggregation level accurately.

In the realm of both classification and segmentation tasks, the range of target object categories spans from a few to several dozen. Consequently, the generation of Class Activation Maps (CAM) for all corresponding categories introduces a computationally onerous endeavor. Given the pivotal role that feature maps play in the computation of CAM, we introduce Spatial Coherence (SC) metric. This metric directly scrutinizes the distributional characteristics of attention regions across all output feature maps within a specific layer, offering insight into the extent of redundancies amenable to pruning in said layer.

We focus on individual feature maps denoted as $X_{i,j} \in \mathbb{R}^{h_i \times w_i}$, where these feature maps represent the output corresponding to the j-th channel of the i-th layer. To ensure fairness, we adjust $X_{i,j}$ using bilinear interpolation to match the size of the input image, resulting in $\widetilde{X}_{i,j} \in \mathbb{R}^{H \times W}$. The threshold, $T_{i,j}$, is computed as the mean of all pixel values within $\widetilde{X}_{i,j}$. Pixels surpassing $T_{i,j}$ are designated as feature attention points with a value of 1, while those equal to or below the threshold are set to 0. Through this process, we obtained $G_{i,j}$, a binary matrix that reflects the distribution status of the attention points of the feature map:

$$G_{i,j,h,w} = \begin{cases} 1 & \text{if } \widetilde{X}_{i,j,h,w} > T_{i,j} \\ 0 & \text{if } \widetilde{X}_{i,j,h,w} \leq T_{i,j} \end{cases} \tag{5}$$

where h and w denote the indices of the last two dimensions of $G_{i,j}$. The attention points of different feature maps exhibit distinct distribution characteristics. To quantify the spatial aggregation level of $G_{i,j}$, we traverse it starting from the top-left corner in a row-major (or column-major) order. Initially, connected components labeling is applied to the attention state map $G_{i,j}$. For each foreground pixel (with a value of 1), if the current pixel is unmarked (label 0), either a depth-first search or breadth-first search algorithm is employed to label all connected foreground pixels, assigning them a common label as a connected region. Subsequently, statistical information is collected for each connected region as shown in Appendix, including the collection of all central coordinates $C = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ within each connected region. Here, n refers to the total number of connected regions. Additionally, essential statistical information such as the total pixel count N within all connected regions is obtained. First, calculate the average distance $\overline{d}$ between search pair of elements in the set of central coordinates C:

$$\bar{d} = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6}$$

Calculate the average pixel count for all connected regions on a feature map: $\bar{N} = N/n$. Next, compute the average metric of all n_{i+1} output feature maps for the i-th layer to reflect the layer's spatial coherence metric:

$$sc_i = \frac{\sum \bar{d} \times \bar{N}}{n_{i+1}} \tag{7}$$

By calculating the sc for each layer, we obtain an m-dimensional vector $\mathrm{SC} = [sc_1, sc_2, \ldots, sc_m]^T$ with the same shape as HM. Larger SC values in the convolutional layers indicate that the attention points (i.e., pixels with a value of 1 in $G_{i,j}$) in the output feature maps are more concentrated in specific regions. This implies that the current layer exhibits sparsity in spatial contextual representation. Therefore, it can be represented with fewer channels.

3.3 Integration of HM and SC

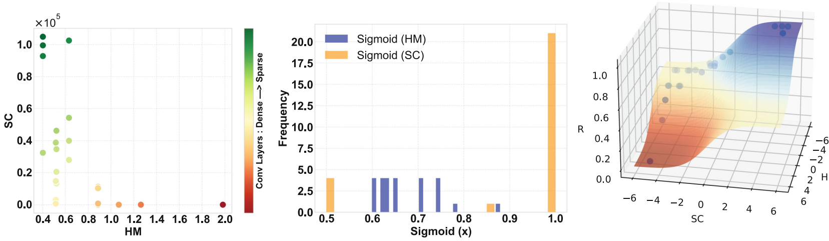
The m-dimensional vectors HM and SC, obtained in our previous discussion, exclusively capture the relative sensitivity and spatial coherence of each layer. It is crucial to integrate the impact of HM and SC on the pruning results in order to arrive at the final pruning strategy. Figure 3(a) depicts the numerical distribution of HM and SC vectors for STDC, revealing substantial differences in their distribution regions and spans. Therefore, our initial step involves independently normalizing these vectors. For each element hm_i in the HM vector and sc_i in the SC vector, we apply min-max normalization to obtain $\hat{HM}$ and $\hat{SC}$:

$$\hat{hm}_i = -s + \frac{2s \cdot (hm_i - \min(HM))}{\max(HM) - \min(HM)} \tag{8}$$

$$\hat{sc}_i = -s + \frac{2s \cdot (sc_i - \min(SC))}{\max(SC) - \min(SC)} \tag{9}$$

where s represents the specified scaling factor to normalize the numerical values of the two vectors within the range $[-s, s]$. Through distribution analysis of the $\hat{HM}$ and $\hat{SC}$ vectors, relative to the fluctuations in intermediate values, the impact of extreme values on the final result is almost negligible. Figure 3(b) illustrates the specific distribution of $\hat{HM}$ and $\hat{SC}$ values on a variant of the sigmoid curve. We found that the distribution of the indicators is extremely scattered or overly concentrated. To fully leverage the characteristics of the sigmoid function, we set s to 5 in Eq. 8, the specific reasons are explained in Appendix. Based on these numerical observations, we employ a variant of the sigmoid function to blend the numerical values of the two-dimensional vectors, as shown below:

$$r_i = \frac{\sigma(\hat{sc}_i) - \sigma(\hat{hm}_i) + 1}{2} \tag{10}$$



(a) Range of HM and SC (b) Sigmoid mapping (c) Mapping of $\hat{HM}$ $\hat{SC}$ and R

Fig. 3. Illustration of HM and SC vectors for the STDC backbone network. (a) The original data distribution. (b) The numerical distribution by directly mapping data to the sigmoid function. (c) The mapping of normalized $\hat{HM}$ and $\hat{SC}$ to R. Red signifies dense, and blue indicates sparsity in convolutional layers (Color figure online)

In the given context, r_i represents an element in R, where $R = (r_1, r_2, \ldots, r_m) \in [0, 1)^m$ denotes the final layer-wise pruning ratios configuration scheme. $\sigma(\ldots)$ corresponds to the sigmoid function, a pivotal mathematical operation frequently employed in neural network computations. Figure 3(c) illustrates a three-dimensional visualization of the mapping from $\hat{HM}$ and $\hat{SC}$ to R. To achieve arbitrary global pruning rates, R can be scaled proportionally and pruning can be constrained within upper and lower bounds.

3.4 Scalability for Residual Network Structures

The above methodology is designed for single-branch networks originally. For networks with residual structures such as ResNet and STDC, we have also employed corresponding pruning methods. We employ the greedy pruning strategy and processing method for ResNet residual modules outlined in [17], and we have devised a specialized pruning strategy for the STDC module introduced in [6], as illustrated in Fig. 4.

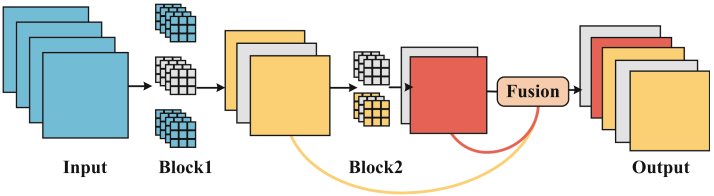


Fig. 4. Pruning process for STDC network architecture. Taking one stage of the STDC network as an example, for clarity, we only depict 2 blocks. The gray areas represent pruned portions

4 Experiments

In our experiments, we compare two pruning ratios configuration schemes: "Identical", where each layer is pruned with the same ratio, and "LPRA", our Layer-wise Pruning Ratios Auto-configuration method. For intra-layer pruning, we consider six methods: L1 [17], FPGM [12], Apoz [13], Taylor [22], Thine [20], and Random. The Random method randomly selects kernels within a layer for removal based on pre-defined ratios, can serving as a baseline. Our experiments aim to compare the effectiveness of foundational pruning strategies and validate the advantages of our LPRA-based automatic pruning ratios determination when using consistent intra-layer pruning strategies.

We conduct experiments on both classification and real-time semantic segmentation task. For classification, we utilize two prominent datasets, CIFAR-10/100 [15] and ImageNet2012 [24]. The experiments encompass two key network architectures: ResNet [10] and VGG [25]. In the ResNet series, pruning is applied selectively, targeting the first convolution layer of normal residual blocks and the first and second convolution layers of bottleneck blocks. In the VGG series, all internal convolutional layers are pruned. Evaluation metrics include Top-1 and Top-5 accuracy, FLOPs, and parameters(Params). For real-time semantic segmentation task, Cityscapes [2] is employed, and pruning is executed on the lightweight STDC network [6]. Figure 4 depicts a pruning strategy specifically designed for the unique structure of STDC. Evaluation metrics for this task encompass mIoU (mean Intersection over Union), Flops, and Params.

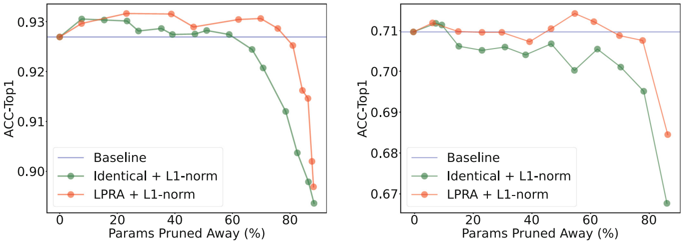
4.1 Experiments on CIFAR-10 and CIFAR-100

The CIFAR-10 and CIFAR-100 datasets contain 50,000 training images and 10,000 test images each, standardized to 32 × 32 pixels. CIFAR-100 covers 100 categories, while CIFAR-10 covers 10 categories. By employing "Identical" and "LPRA" pruning schemes, along with six intra-layer pruning strategies, we compressed ResNet and VGG architectures, achieving comparable reductions in both FLOPs and Params. In particular, only 10 randomly selected batches of the training set served as data dependencies for LPRA. Following pruning, the networks underwent 300 epochs of fine-tuning with parameters aligned to the training regimen.

We depict the correlation between Top-1 and Top-5 accuracy changes and the pruning ratios of FLOPs and parameters. Table 1 and Table 2 depict the results for VGG-16 and ResNet-34, other experimental results are shown in Appendix. The results indicate that intra-layer pruning schemes minimally impact accuracy, as even random pruning attains levels akin to more intricate methods. In contrast, intra-layer pruning ratios guided by our LPRA strategy, demonstrate noticeably higher accuracy. Notably, LPRA achieves a significantly larger Params compression ratio at similar FLOPs compression ratio compared to the Identical scheme. Additionally, Fig. 5 illustrates accuracy change curves for ResNet-18 networks compressed at various ratios under both Identical and LPRA schemes, Here, the fine-tuning epoch is set to 60, with an initial learning rate of

Table 1. Retain 60% of FLOPs for VGG-16 on CIFAR-100. "Baseline" refers to the data of the original model. For Ratios: "Identical" means using the same pruning rate for each layer, "LPRA" is our pruning rate configuration method. "Crit." indicates the specific pruning strategy within each layer.

Ratios	Crit.	Top-1 Acc(%)	Top-5 Acc(%)	Params[M]/Ratio(%)
VGG-16, CIFAR-100, Target FLOPs Ratio 60%				
Baseline		72.80	91.32	14.77/100.0
Identical	L1	72.07(−0.73)	90.84(−0.48)	8.86/59.99
	FPGM	72.79(−0.01)	91.13(−0.19)	
	Apoz	71.75(−1.05)	91.10(−0.22)	
	Taylor	72.10(−0.70)	91.19(−0.13)	
	Thinet	72.27(−0.53)	90.78(−0.54)	
	Random	72.19(−0.61)	91.02(−0.30)	
LPRA	L1	73.03(+0.23)	91.27(−0.05)	**7.40/50.10**
	FPGM	72.63(−0.17)	91.37(+0.05)	
	Apoz	**73.16(+0.36)**	91.30(−0.02)	
	Taylor	72.77(−0.03)	91.29(−0.03)	
	Thinet	72.66(−0.14)	**91.46(+0.14)**	
	Random	72.53(−0.27)	91.36(+0.04)	



(a) (b)

Fig. 5. Top-1 accuracy obtained by pruning ResNet-18 with different pruning rates, where (a) represents the results on CIFAR-10, (b) represents the results on CIFAR-100. The blue line indicates the accuracy before pruning (Color figure online)

0.1, decreasing to 0.1 times the original rate every 20 epochs. We found that the LPRA method can maintain or even surpass the accuracy of the original model even when pruning 70% of the Params.

Table 2. The results of preserving 70% of FLOPs for the ResNet-34 network on CIFAR-10.

Ratios	Crit.	Top-1 Acc(%)	Top-5 Acc(%)	Params[M]/Ratio(%)
ResNet-34, CIFAR-10, Target FLOPs Ratio 70%				
Baseline		93.57	99.77	21.28/100.0
Identical	L1	93.60(+0.03)	99.80(+0.03)	14.76/69.36
	FPGM	93.39(−0.18)	99.75(−0.02)	
	Apoz	93.35(−0.22)	99.81(+0.04)	
	Taylor	93.62(+0.05)	99.81(+0.04)	
	Thinet	93.55(−0.02)	99.74(−0.03)	
	Random	93.41(−0.16)	99.75(−0.02)	
LPRA	L1	**93.73(+0.16)**	99.82(+0.05)	**12.50/58.74**
	FPGM	93.46(−0.11)	99.83(+0.06)	
	Apoz	93.70(+0.13)	**99.86(+0.09)**	
	Taylor	93.59(+0.02)	99.79(+0.02)	
	Thinet	93.57(+0.00)	99.77(+0.00)	
	Random	93.49(−0.08)	99.78(+0.01)	

4.2 Experiments on ImageNet

ImageNet2012 comprises 1.28 million annotated images, with 1.2 million utilized for training, and the remaining 50,000 divided into validation and test sets. Each image has a resolution of 224 × 224 pixels. Unlike Sect. 4.1, due to the large size of the ImageNet dataset and to save computation time, we set the fine-tuning epochs to 20, with an initial learning rate of 1×10^{-2}, decreasing to 0.1 times the original rate every 10 epochs. Table 3 presents the pruning results of ResNet-50 on ImageNet, demonstrating the continued effectiveness of LPRA on large datasets.

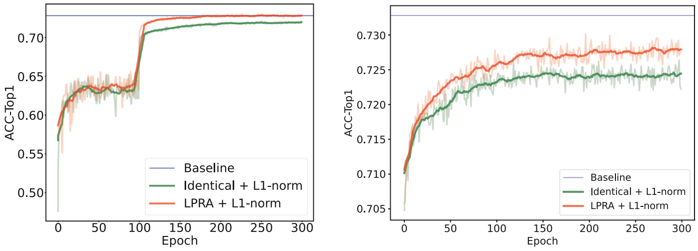
4.3 Experiments on STDC Network

The Cityscapes dataset contains 5000 annotated images, with 2975 used for training, 500 for validation, and 1525 for testing. Each image has a resolution of 1024 × 2048. Unlike image classification tasks, for improved computational efficiency, we randomly selected 5 images from the training set as the basis for LPRA calculation. This was done to reduce similar FLOPs and Params in the STDC network. After pruning, the learning rate was reduced to one hundredth of the training learning rate, followed by fine-tuning for 30 epochs.

Table 4 presents the pruning results of the STDC-75 network with an input image size of 0.75 × (512 × 1024). In cases with similar FLOPs and Params, the LPRA achieved significantly higher accuracy compared to the Identical scheme. Notably, under the LPRA pruning rate configuration scheme, the Random pruning strategy outperformed most pruning strategies under the Identical scheme.

Table 3. The results of preserving 67.44% of FLOPs for the ResNet-34 network on ImageNet2012

Ratios	Crit.	Top-1 Acc(%)	Top-5 Acc(%)	Params[M]/Ratio(%)
ResNet-50, ImageNet2012, Target FLOPs Ratio 67.44%				
Baseline		76.13	92.86	25.56/100.0
Identical	L1	74.29(−1.84)	92.14(−0.72)	17.47/68.35
	FPGM	73.72(−2.14)	91.93(−0.93)	
	Apoz	74.02(−2.11)	92.00(−0.86)	
	Taylor	74.22(−1.91)	92.03(−0.83)	
	Thinet	74.27(−1.86)	92.07(−0.79)	
	Random	74.20(−1.93)	92.00(−0.86)	
LPRA	L1	**74.54(–1.59)**	92.21(−0.65)	**16.62/65.02**
	FPGM	74.05(−2.08)	92.05(−0.81)	
	Apoz	74.46(−1.67)	**92.12(–0.74)**	
	Taylor	74.29(−1.84)	92.03(−0.83)	
	Thinet	73.86(−2.27)	91.92(−0.94)	
	Random	73.86(−2.27)	91.90(−0.96)	



(a) VGG-16

(b) Resnet-34

Fig. 6. Relationship between fine-tuning epochs and Top-1 accuracy after pruning 40% of FLOPs from VGG-16 and 32% of FLOPs from ResNet-34 on CIFAR-100

This further underscores the effectiveness of our LPRA pruning rate automatic configuration scheme.

4.4 Ablation Studies

In this section, we will explore the impact of the number of fine-tuning epochs on the final accuracy and attempt an ablation study on the main components of LPRA.

The Impact of Retraining-Epoches. To investigate the impact of fine-tuning iterations on the final accuracy, we extensively fine-tuned the pruned models, recording accuracy changes over 300 iterations. Figure 6 illustrates results for pruning 40% of VGG-16 on CIFAR100. Results from other networks and datasets are presented in Appendix. The red line represents the LPRA scheme, and the green line represents the Identical scheme, both using L1 norm for intra-layer pruning. The initial learning rate is set to 0.1, reducing to 0.1 times the original value every 100 epoches. Visual results show that as fine-tuning iterations increase, the model's accuracy on the validation set generally rises. However, there is an apparent upper limit: after a certain number of iterations, the model's accuracy ceases to significantly improve.

The Impact of HM. We use the HM vector of the convolution layer to represent its sensitivity and the SC vector to represent the spatial context information representation ability. In order to explore the effect of Hessian Mean (HM) analysis (corresponding to the HM analysis module placed on Fig. 1) on the pruning accuracy of the pruned model, we keep only the Spatial Coherence (SC) part below Fig. 1 to obtain the SC vector. We then use the min-max normalization module and the Nonlinear map to process SC and obtain the final pruning ratios R. Table 3 shows the experimental results of VGG-16 on CIFAR-100 and ResNet-34 on CIFAR-10. We observe that removing HM analysis results in a slight accuracy decrease, while the compression of FLOPs is significantly reduced, indicating its crucial role in maintaining model predictive performance and reducing FLOPs.

Table 4. Pruning Results of the STDC Semantic Segmentation Network on Cityscapes

Ratios	Crit.	mIoU(%)	FLOPs[G]/Ratio(%)	Params[M]/Ratio(%)
Baseline		74.13	35.34/100.0	14.24/100.0
Identical	L1	70.38(−3.75)	21.16/59.88	8.30/58.29
	FPGM	69.87(−4.26)	21.84/61.80	8.52/59.83
	Apoz	68.62(−5.51)	21.16/59.88	8.30/58.29
	Taylor	69.19(−4.94)	21.16/59.88	8.30/58.29
	Thinet	67.15(−6.98)	**21.09/59.68**	8.29/58.22
	Random	67.50(−6.63)	21.16/59.72	8.30/58.29
LPRA	L1	**71.08(−3.05)**	21.84/61.80	8.25/57.94
	FPGM	70.54(−3.59)	24.55/69.47	**8.23/57.79**
	Apoz	70.14(−3.99)	21.45/60.70	8.28/58.15
	Taylor	71.01(−3.12)	21.45/60.70	8.28/58.15
	Thinet	69.08(−5.05)	21.26/60.16	8.37/58.78
	Random	70.67(−3.46)	21.45/60.70	8.28/58.15

The Impact of SC. *Similarly*, in order to explore the effectiveness of Spatial Coherence (SC) analysis, we removed the SC analysis module and directly used the min-max normalization module and Nonlinear map to process the HM vector to obtain the final pruning ratios R. From the results in Table 5, we found that

Table 5. The impact of HM and SC modules on pruning results on VGG-16 and ResNet-34. "w/o HM" means removing HM analysis, "w/o SC" means removing SC analysis.

Ratios	Top-1 Acc (%)	FLOPs[G]/Ratio(%)	Params[M]/Ratio(%)
VGG-16 CIFAR-100 Crit.L1			
Baseline	72.80	0.39/100.0	14.77/100.0
LPRA	**71.20(−1.60)**	0.15/38.46	3.44/23.29
w/o HM	71.16(−1.64)	0.19/48.72	3.46/23.43
w/o SC	70.74(−2.06)	0.15/38.46	3.98/26.95
Identical	70.15(−2.65)	0.15/38.46	5.65/38.25
ResNet-34 CIFAR-10 Crit.L1			
Baseline	93.57	0.99/100.0	21.28/100.0
LPRA	**93.47(−0.10)**	0.57/57.58	8.81/41.40
w/o HM	93.45(−0.12)	0.76/76.77	9.03/42.43
w/o SC	93.27(−0.30)	0.55/55.56	11.17/52.49
Identical	93.38(−0.19)	0.53/53.54	11.38/53.48

compared to the LPRA scheme, removing SC leads to a significant decrease in the model's accuracy, indicating that SC analysis plays a crucial role in maintaining accuracy and compressing Params.

5 Conclusion

This paper introduces LPRA, an end-to-end pruning ratios configuration framework, benchmarking various structured pruning techniques. With limited data and computation, we employ sensitivity and spatial context analysis to quantify sparsity in each convolutional layer, with the aim of preserving the network's inference capability after pruning. Extensive experimental results demonstrate the competitive performance of LPRA across various network architectures and datasets compared to commonly used pruning ratios configuration methods. Particularly noteworthy is its commendable outcomes in parameter compression and accuracy retention. Due to the flexibility and universality of LPRA, it can be readily integrated with other methods to achieve enhanced model compression results.

References

1. Chen, Z., Chen, Z., Lin, J., Liu, S., Li, W.: Deep neural network acceleration based on low-rank approximated channel pruning. IEEE Trans. Circuits Syst. I Regul. Pap. **67**(4), 1232–1244 (2020)
2. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)

3. Ding, X., et al.: Resrep: lossless CNN pruning via decoupling remembering and forgetting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4510–4520 (2021)
4. Dong, Z., Yao, Z., Arfeen, D., Gholami, A., Mahoney, M.W., Keutzer, K.: HAWQ-V2: hessian aware trace-weighted quantization of neural networks. In: Advances in Neural Information Processing Systems, vol. 33, pp. 18518–18529 (2020)
5. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: HAWQ: hessian aware quantization of neural networks with mixed-precision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 293–302 (2019)
6. Fan, M., et al.: Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9716–9725 (2021)
7. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: a survey. Int. J. Comput. Vis. **129**, 1789–1819 (2021)
8. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
9. Hassibi, B., Stork, D.: Second order derivatives for network pruning: optimal brain surgeon. In: Advances in Neural Information Processing Systems, vol. 5 (1992)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. IEEE (2016)
11. He, W., Wu, M., Liang, M., Lam, S.K.: Cap: context-aware pruning for semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 960–969 (2021)
12. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4340–4349 (2019)
13. Hu, H., Peng, R., Tai, Y.W., Tang, C.K.: Network trimming: a data-driven neuron pruning approach towards efficient deep architectures. arXiv preprint arXiv:1607.03250 (2016)
14. Jacob, B., et al.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2704–2713 (2018)
15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
16. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In: Advances in Neural Information Processing Systems, vol. 2 (1989)
17. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. In: International Conference on Learning Representations (ICLR) (2017)
18. Li, Y., Adamczewski, K., Li, W., Gu, S., Timofte, R., Van Gool, L.: Revisiting random channel pruning for neural network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 191–201 (2022)
19. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2736–2744 (2017)
20. Luo, J.H., Wu, J., Lin, W.: Thinet: a filter level pruning method for deep neural network compression. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5058–5066 (2017)

21. Molchanov, D., Ashukha, A., Vetrov, D.: Variational dropout sparsifies deep neural networks. In: International Conference on Machine Learning, pp. 2498–2507. PMLR (2017)
22. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J.: Importance estimation for neural network pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11264–11272 (2019)
23. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3967–3976 (2019)
24. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. (2014)
26. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: HAQ: hardware-aware automated quantization with mixed precision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8612–8620 (2019)
27. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
28. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
29. Zhuang, Z., et al.: Discrimination-aware channel pruning for deep neural networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)

Hyperspectral Imaging for Characterization of Construction Waste Material in Recycling Applications

Hannah Frank[1(✉)], Karl Vetter[1], Leon A. Varga[1], Lars Wolff[2], and Andreas Zell[1]

[1] Cognitive Systems, University of Tuebingen, Tuebingen, Germany
{hannah.frank,karl.vetter,leon-amadeus.varga, andreas.zell}@uni-tuebingen.de
[2] Optocycle GmbH, Tuebingen, Germany
l.wolff@optocycle.com

Abstract. Hyperspectral imaging offers a way for computer vision to surpass human visual perception by increasing the available spectral information beyond RGB images. The requirement of specific cameras and difficulty of capturing hyperspectral images has led to a scarcity of data, with most belonging to the aerial remote sensing paradigm. In this paper, we present a novel and extensive hyperspectral dataset of construction debris for recycling applications that includes objects from 14 different material classes, measured by three different hyperspectral sensors. We compare a variety of hyperspectral image classification approaches and demonstrate that relative performance of common hyperspectral models differs for our new dataset. Furthermore, we demonstrate a positive effect of pre-training on our dataset for other similar hyperspectral image classification tasks.

Keywords: Hyperspectral Imaging · Dataset · Classification

1 Introduction

In recent years, computer vision algorithms have undergone significant improvements, rivaling human performance. Most applications limit their algorithms by providing only three color channels in the visible spectral range. Hyperspectral imaging (HSI) overcomes this limitation by recording up to hundreds of wavelengths beyond the visible light, especially in the near-infrared (NIR) range, and thus offers a way for computer vision to surpass human visual perception.

Its utility in tasks where spectral information is important has made it increasingly popular and nowadays, it is applied in many different fields, like in remote sensing where HSI has its origin (e.g., [7]). But also in medical applications [20], in agriculture [19], in the food industry [4,28], and in the recycling sector where the objective is, for example, the sorting of plastic or construction and demolition waste (CDW) material [3,25].

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 160–175, 2025.
https://doi.org/10.1007/978-3-031-78444-6_11

The latter has become particularly relevant as environmental protection comes to the forefront of public policymaking, and interest in avenues of decreasing waste has grown. Nowadays, CDW makes up almost one fourth of globally produced waste, and millions of tons of concrete waste are generated annually, most of which could be recycled. However, recycling is only possible if the reusable material can be identified and effectively separated from other, contaminating material. At the waste processing facility where hundreds of trucks loads of CDW come in daily, an automatic classification of the material is required.

Various approaches and methods for classifying hyperspectral data have been developed over the years. While in the early stages, classical machine learning approaches, like support vector machines (SVM) [31], were used on a pixel-wise basis, nowadays, hyperspectral image data is mostly evaluated using more sophisticated approaches that are able to incorporate spatial context in addition to the spectral information. These include convolutional neural networks (CNN) of different complexity [6,23,24], and most recently also vision transformers (ViT) [12,33] that are currently achieving outstanding results in all kinds of computer vision tasks.

Ideally, these classification models should be as generic and generally applicable to HSI data as possible. However, the application of construction waste sorting, for instance, is completely different from the task of segmenting areas in remote sensing. Still, design and evaluation of HSI classification methods happens almost exclusively based on a small number of remote sensing scenes (e.g., [23]). A recurring challenge in the field of HSI lies in the limited availability and size of hyperspectral datasets, impeding exhaustive architecture search for more generally suitable classifier models. Consequently, these are the main contributions of our work:

- We present a novel and completely annotated hyperspectral dataset, consisting of recordings of 14 different CDW material classes, measured by three different sensors over more than 200 bands in the visible and near-infrared range.
- Based on this dataset that enables object-wise classification as well as pixel- or patch-wise prediction, we present a comprehensive evaluation and comparison of multiple HSI classification methods. Further, we compare classifier performance in two other applications – where we observe that methods perform indeed different on this dataset relative to existing ones.
- Additionally, we show that pre-training the HSI classifiers on our dataset can also be helpful for other, similar classification tasks.

2 Related Work

Hyperspectral Datasets. With the hyperspectral imaging technique becoming more popular, there already exist many publications on measuring and processing this particular type of recording with many spectral bands. However, only a few corresponding datasets are actually publicly available.

Firstly, worth mentioning are the *CAVE* dataset [34] as well as the *Havard* hyperspectral "real world" dataset [5], although they both only provide 31 spectral bands – making them rather multispectral than hyperspectral image datasets.

Actual hyperspectral datasets stem mainly from remote sensing applications, like the most popular and frequently used *Hyperspectral Remote Sensing Scenes* (HRSS) [9], but there are some other similar datasets [1,13,14,22]. There is data from the agricultural sector [16,21] as well as from the food industry, depicting pasta [4], or fruit, like for example the *DeepHS Fruit* and *DeepHS Fruit v2* dataset [27,28]. Then, there is also some medical data [26]. In the context of material characterization, we can find small public datasets covering, e.g., plastic types [18], or components of building facades [10]. A freely available dataset that includes CDW material for application in waste sorting and recycling does – to the best of our knowledge – not yet exist.

Hyperspectral Image Classification. To classify hyperspectral images, in the early stages, classical ML approaches, like support vector machines (SVM) [31] or partial least squares analysis (PLS-DA) [3,25] were used, often in combination with feature extraction or dimensionality reduction, e.g., via principal component analysis (PCA), as a preprocessing step. These methods operate pixel-wise and therefore on the spectral dimension only.

Nowadays, hyperspectral image data is mostly evaluated using deep learning. Deep learning models, such as autoencoders, recurrent neural networks and especially convolutional neural networks (CNN), have been successfully applied for HSI classification [6,23,24].

Conventional CNNs use 2D convolutions and operate on patches of the image or even on the entire image, utilizing the spatial information. However, it has been shown that HSI classification performance is highly dependent on both spatial and spectral information [23]. 3D convolutions can act on the spectral and spatial dimension simultaneously, but at the cost of increased computational complexity. Most recently, there have been attempts to combine both 2D and 3D convolutions in order to benefit from the spatial and spectral feature learning capability, by simultaneously overcoming the latter drawback. Such networks, like the one proposed by Roy et al. [24], are then referred to as hybrid CNNs or 3D-2D CNNs. An alternative approach is to apply spectral transformations and then again use 2D convolutions on the transformed data. For instance, Chakraborty et al. [6] employed wavelet transformations to incorporate both aspects.

Although CNNs have proven to be powerful in extracting spatial and locally contextual information, they fail to capture the global information, especially long-term dependencies in the spectral dimension of the hyperspectral data. Vision transformers (ViT) are specifically designed for analyzing sequential data and accounting for global information, and therefore have also been successfully applied for HSI classification recently [12,33].

The current development of HSI models has been reviewed by a couple of publications already [2,23,30]. However, these reviews often lack the most recent developments, including vision transformers, and focus mostly on remote sensing scenes.

Pre-training on Hyperspectral Image Data. Most classifier models highly depend on the amount of training data available, but only a small number of annotated samples is usually available from HS datasets for a specific application. A potential workaround is to pre-train the classifier model on other available data of similar structure, and then just fine-tune it on the actual application scenario.

As this has been shown to stabilize training, avoid overfitting, improve classification accuracy and allow for the use of larger (deeper) models in general, pre-training is nowadays state-of-the-art for RGB data. It has also already been explored for hyperspectral image classification to some extent [17,32]. Lee et al. [17], for example, showed that pre-training works on hyperspectral data as well, and that it is even possible to pre-train a shared backbone using a multi-domain approach. However, they only considered different remote sensing scenes, with both data and task still being very similar across those domains.

In contrast, we conduct pre-training on our *DeepHS Debris* dataset and ask whether general hyperspectral features can be learned and subsequently transferred to entirely different application scenarios, like remote sensing or food inspection where data, recorded wavelengths and classification task vary considerably.

Application: HSI in Waste Management and Recycling. There is already some existing research on the application of hyperspectral imaging in the waste management and recycling sector, esp. work on CDW material classification. Generally, the datasets used in these works are extremely small, and none of them is – to the best of our knowledge – publicly available. Further, mostly very basic ML approaches were used, completely ignoring the potential of promising, more recent approaches using deep learning (e.g., CNNs or ViTs).

For example, Serranti et al. [25] and Bonifazi et al. [3] investigated the use of hyperspectral images for recycling and successfully performed classification of both construction waste and different plastic types. However, these results were achieved exclusively applying PLS-DA, a very basic ML approach, and using only a few material types as part of a very small, unpublished dataset.

3 Dataset: *DeepHS Debris*

We present a new hyperspectral dataset that depicts objects of different CDW material classes. The data was recorded under laboratory conditions, using three hyperspectral line-scan cameras (Corning microHSI 410 Vis-NIR Hyperspectral Sensor, Innospec RedEye 1.7, Specim FX10), covering the visible and the lower near-infrared range (from 400 nm up to 1700 nm) and producing high-resolution

Table 1. List of the hyperspectral cameras and their specifications

Camera	# Bands	Wavelength range	Spatial width
Corning microHSI 410 Sensor	249	408–901 nm	1486 px
Innospec RedEye 1.7	252	920–1730 nm	320 px
Specim FX10	224	400–1000 nm	600 px

images of the debris under consideration. The detailed specifications of the cameras can be found in Table 1.

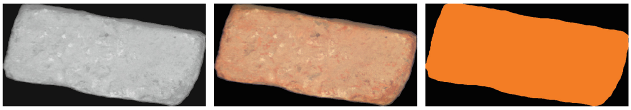
Each camera recording contains a single debris object, belonging to one of 14 material classes. For a detailed breakdown of the classes, refer to Table 2. As each of the 215 objects was recorded from two sides, using at least two different hyperspectral cameras, this results in overall 860 recordings.

Table 2. List and number of recorded objects and corresponding hyperspectral recordings for the material classes covered by the *DeepHS Debris* dataset

Class name	# Objects	# Recordings
Concrete	35	140
Stone	20	80
Tile	25	100
Brick	25	100
Porous material	10	40
Bituminous material	15	60
Soil	10	40
Sand	10	40
Glass	10	40
Ceramic	15	60
Wood	10	40
Plastic	10	40
Metal	10	40
Paper	10	40
	$\sum$ 215	$\sum$ 860

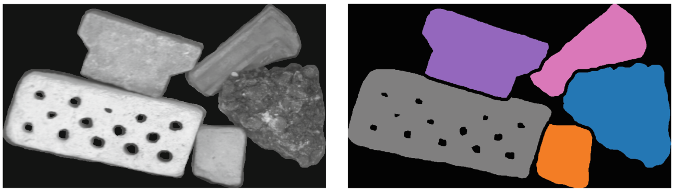
The raw recordings were subjected to the following pre-processing: Referencing using a white and dark reference recorded at the beginning of every measurement, background extraction and cropping. Finally, each recording was resized to 512 px longer edge, keeping the ratio.

For all recordings, we provide both global object-wise and pixel-wise annotation – in the form of a class label and a color-coded segmentation mask, respectively. Figure 1 shows an annotated sample recording of a brick, recorded by the



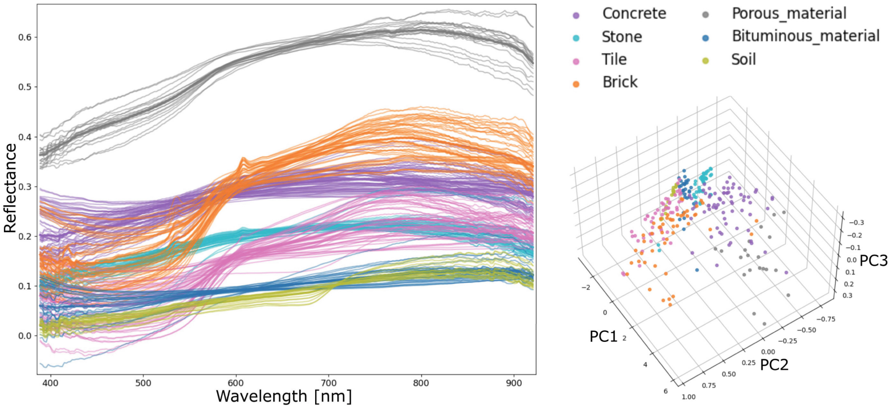
(a) (b) (c)

Fig. 1. A brick recording of the *DeepHS Debris* dataset. (a) depicts the average of the bands. (b) shows a reconstructed RGB image, (c) shows the corresponding ground-truth segmentation mask. An orange colored pixel represents the class brick. The global ground-truth is also "brick", correspondingly (Color figure online)



(a) (b)

Fig. 2. A recording of multiple materials at once. (a) depicts the average of the bands, and (b) shows the corresponding segmentation mask. The recording contains concrete (purple), porous material (gray), brick (orange), tile (pink), and bituminous material (blue) (Color figure online)



(a) Characteristic spectra (b) Visualization after PCA

Fig. 3. Visual analysis of the spectral information in the data: (a) shows the characteristic spectra, with the respective class mean in bold, and in (b), the projection to the first three principal components over the spectral dimension (mean over pixels per sample) is plotted, for the Corning HSI camera and for the most frequent material classes: Concrete (purple), tile (pink), brick (orange), porous material (gray), bituminous material (blue), stone (turquoise) and soil (light green) (Color figure online)

Corning camera. Also, some combinations of multiple debris of different material classes were recorded, and we provide the corresponding class segmentation masks. An example is shown in Fig. 2.

In addition to the plain collection of hyperspectral recordings itself, we provide some fundamental spectral analysis on our dataset: Fig. 3a shows the characteristic spectra curves for a selection of the most frequently found CDW material classes. Further, we employed PCA for dimensionality reduction – again over the spectral dimension – and show the samples' projection on the first three principal components in Fig. 3b.

Both visualizations indicate that distinction between material classes based on the spectral information contained in a hyperspectral recording is possible. However, it is not as straightforward as one may have expected. A clear separation in low-dimensional space is not possible, and the fact that the class-specific spectra curves show areas of overlap and variance in-between samples, confirm the need for more sophisticated approaches and models for classification.

4 Experiments

4.1 Methods

We utilize our dataset to re-evaluate a number of methods, based on fundamentally different approaches to handle the hyperspectral data during classification.

The first category comprises classical machine learning techniques operating on a pixel-basis, namely SVM with an RBF kernel (similar to Waske et al. [31]), and PLS-DA. Secondly, we evaluate basic CNNs with kernels of different dimensionality: Similar to Paoletti et al. [23], we use a 1D CNN that employs 1D convolution layers only along the spectral dimension of each pixel. a 2D CNN that convolves the input (image patches or whole images) in the spatial dimension using 2D convolutional layers, while combining the spectral data in the fully connected head, and finally, a 3D CNN that is able to incorporate all three dimensions of the hyperspectral cube at the same time. Furthermore, we evaluate a spatial 2D CNN solely focusing on spatial features by "squashing" the spectral dimension averaging over all channels, or applying a Gabor filter enhancing textural information in case of the Gabor CNN [8] beforehand. Another 2D CNN-based architecture, but with shortcut connections enhancing inter-connectivity between layers at varying depths, is the ResNet proposed by [11]. For our experiments, we utilize the ResNet-18 and a ResNet-152, representing a larger-scale CNN architecture. DeepHS-Net [28] is again a 2D CNN, specifically designed for efficient performance on small hyperspectral datasets, like *DeepHS Fruit* [27,28]. DeepHS-Hybrid-Net [27], adds a layer of 3D convolutions – leveraging both spectral and spatial dimension and simultaneously reducing the number of parameters as compared to a fully 3D CNN. Another improved variant of the DeepHS-Net replaces the first layer by a HyveConv layer [29]. We further evaluate two state-of-the-art methods for remote sensing: SpectralNET [6], which utilizes wavelet transformations to conduct convolutions in both the spatial and spectral dimensions of image patches, and HybridSN [24] which again combines regular 3D and 2D convolutions. The last category represents transformer models, adapted for hyperspectral image classification: SpectralFormer, introduced by Hong et al. [12], and the HiT [33] which is a ViT model including 3D convolution projection modules and convolution permutators to capture subtle spatial-spectral discrepancies, both again operating on image patches.

Based on the complexity of the model and/or suggestions by the authors, some models are applied to a PCA-reduced or otherwise preprocessed input.

An overview of the individual methods and additional information can be found in Table 3.

Table 3. Overview and further details on the methods

Model	Type	# Param.	Spectral inform.	Spatial context	Input	Pre-process.
SVM	Classic ML	-	✓	✗	Pixel	PCA(10)
PLS-DA		-	✓	✗	Pixel	Raw
1D CNN [23]	CNN	73,000	✓	✗	Pixel	Raw
2D CNN [23]		7,700,000	✓	✓	Patch/obj.	PCA(40)
2D CNN (spatial)		7,400,000	✗	✓	Patch/obj.	Mean
Gabor CNN [8]		7,400,000	✗	✓	Patch/obj.	PCA(3)
3D CNN [23]		44,000,000	✓	✓	Patch/obj.	PCA(40)
ResNet-18 [11]	CNN	12,000,000	✓	✓	Patch/obj.	Raw
ResNet-152 [11]		59,000,000	✓	✓	Patch/obj.	Raw
DeepHS-Net [28]	CNN	31,000	✓	✓	Patch/obj.	Raw
DeepHS-Net+HyveConv [29]		17,000	✓	✓	Patch/obj.	Raw
DeepHS-Hybrid-Net [27]		210,000	✓	✓	Patch/obj.	PCA(40)
SpectralNET [6]	CNN	8,300,000	✓	✓	Patch	Raw
HybridSN [24]		50,000,000	✓	✓	Patch/obj.	PCA(30)
SpectralFormer [12]	ViT	1,100,000	✓	✓	Patch	Raw
HiT [33]		59,000,000	✓	✓	Patch	Raw

4.2 Training Procedure and Evaluation

In order to enable a fair comparison between all the methods introduced in Sect. 4.1, the training procedure was homogenized as far as possible.

The model parameters were optimized with Adam [15], using a learning rate of 0.01, which was stepwise decreased during training. Cross-entropy loss was used as loss function. We trained for 50 epochs and used checkpoint callback and early stopping based on the validation loss. A batch size of 32 was chosen. The training data was augmented using random flipping, random rotation and random cut, each with a probability of 50%, and random cropping with 10% probability. Individual model-specific exceptions exist.

The experiments were primarily conducted on the *DeepHS Debris* dataset of construction waste objects, described in the previous Sect. 3. We additionally included two other, existing HS datasets:

- The *DeepHS Fruit v2* dataset [27,28] containing HS recordings of exotic fruit, that can be categorized into three ripeness levels (unripe, ripe overripe). As this data was recorded at the same chair, (partially) the same measurement settings, esp. sensors (wavelength range 400–1000 nm), were used.

- The widely-used HRSS collection [9], containing remote sensing scenes, recorded by another kind of sensor (up to 2500 nm), and for a completely different task of pixel-wise land cover classification.

Please refer to the original publications for a detailed description and further information.

For all three datasets, we defined a random but fixed train-val-test-split (i.e., 70%–10%–20% for *DeepHS Debris*), and balanced the size of the classes in the categories, respectively. Also, we used a standardized input image size across datasets and models. For object-wise classification, the whole image was resized to 128×128 pixels while for patch-wise classification, we used patches of size 63 pixels, in combination with a dilation of one for the HRSS datasets and 30 for the *DeepHS Debris* dataset, respectively. For testing, all available pixels were used (dilation = 1).

We trained and evaluated each classifier model on the fundamentally different approaches – object-wise and pixel- or patch-wise classification –, if possible. For all combinations, the average accuracy over three different seeds was reported. Although the pixel-/patchwise classification can also be regarded as a segmentation task which would require different performance measures (such as IoU), we decided to stick with (pixel-wise) accuracy for better comparability.

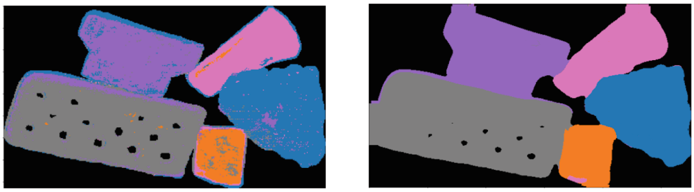
4.3 Pre-training

Aside from evaluating HSI classification approaches, our dataset might further be used for pre-training classifier networks and to meaningfully initialize their weights for subsequent fine-tuning and application in other HSI application scenarios.

For the pre-training experiments, we constrained ourselves to the HyveConv variant of the DeepHS-Net. As the hyperspectral visual embedding convolution (HyveConv) layer [29] operates on a wavelength-based feature learning paradigm, rather than the conventional channel-based approach, it ensures applicability across different camera setups – avoiding the need to employ a separate first layer when transferring to data recorded by another sensor later.

We employed the following pre-training and fine-tuning procedure.

1. **Pre-training** We pre-trained the initial HyveConv network on our *DeepHS Debris* dataset using the same procedure, hyperparameters and augmentations as for regular classifier training (see Sect. 4.2).
2. **Fine-tuning** To transfer to and specialize on another sensor or one of the other aforementioned hyperspectral datasets and corresponding task(s) (see Sect. 4.2), we re-initialized the fully-connected task-specific head (except for the BN layer) to adapt to the differing class outputs, while keeping the pre-trained weights of the remaining layers. The resulting model was again optimized in an end-to-end fashion as described in Sect. 4.2, but only for 30 epochs and with a reduced learning rate of 0.001 – effectively training the last layer from scratch, while only fine-tuning the general backbone part.



(a) SVM (b) DeepHS-Net+HyveConv

Fig. 4. Predicted class mask for a Corning HSI recording of multiple materials: Concrete (purple), porous material (grey), brick (orange), tile (pink), and bituminous material (blue) by (a) SVM, and (b) DeepHS-Net+HyveConv model (Color figure online)

Table 4. Classification accuracy on the *DeepHS Debris* data set (most relevant material classes: concrete, tile, brick, porous material, bituminous material, stone, soil) [in %] for the different classifier models and classification approaches, camera-wise and averaged over the three sensors. The three highest average accuracies object-wise and pixel-/patch-wise are highlighted (bold and underlined)

Model	Approach	Innospec	Corning	Specim	Avg.
	(unit)	RedEye	HSI	FX10	
SVM	Pixel	32.14	61.11	50.00	47.75
PLS-DA	Pixel	50.00	57.14	37.50	48.21
1D CNN	Pixel	63.87	68.86	69.44	67.39
2D CNN	Object	72.14	76.11	77.50	75.25
	Patch	80.25	80.28	63.61	74.71
2D CNN (spatial)	Object	42.86	50.00	50.00	47.62
	Patch	26.69	60.29	60.56	49.18
Gabor CNN	Object	32.14	80.56	87.50	66.73
	Patch	58.02	78.56	68.89	68.49
3D CNN	Object	85.71	83.33	87.50	**85.51**
	Patch	83.27	72.84	66.94	74.35
ResNet-18	Object	35.71	83.33	50.00	56.35
	Patch	74.00	78.77	80.56	77.78
ResNet-152	Object	28.57	61.11	25.00	38.23
	Patch	76.32	76.24	72.22	74.93
DeepHS-Net	Object	57.14	91.67	75.00	74.60
	Patch	57.73	68.48	81.67	69.29
DeepHS-Net+HyveConv	Object	71.43	88.89	75.00	**78.44**
	Patch	71.16	82.22	71.39	74.92
DeepHS-Hybrid-Net	Object	67.86	80.56	77.50	**75.31**
	Patch	69.95	80.93	74.17	75.02
SpectralNET	Patch	69.20	80.56	68.89	72.88
HybridSN	Object	71.43	86.11	62.50	73.35
	Patch	84.60	79.63	70.83	78.35
SpectralFormer	Patch	79.73	81.84	70.56	77.38
HiT	Patch	82.63	79.26	83.33	81.74

5 Results

In this section, we present the results of our previously described experiments.

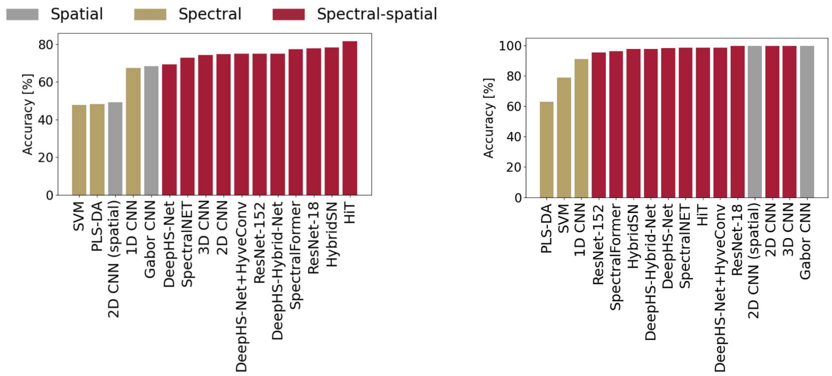
Table 4 lists the classification accuracy for all approaches and sensors, and all classifiers enumerated in Sect. 4.1 when evaluated on a subset of the *DeepHS Debris* data set, containing the seven most frequent and therefore most relevant material classes in practice (concrete, tile, brick, porous material, bituminous material, stone, soil).

Besides the obvious differences in performance between the most simple and more sophisticated deep learning approaches, we also observe substantial disparity in accuracy between pixel-wise, patch-wise and object-wise classification approaches, even for the very same model.

The advantage of incorporating the spatial context with the patch-wise approach in contrast to considering only one pixel at a time, is visible in Fig. 4 containing the predicted class masks for a multi-class recording for the SVM and DeepHS-Net+HyveConv, respectively. The corresponding ground truth can be found in Fig. 2(b).

For patch-wise versus global object-wise classification, although the approaches are not directly comparable, their difference might partially be explained by the varying number of data points available: While for the object-wise classification, the training set size actually equals the (small) number of recordings, for patch-wise classification, the division into multiple small patches results in many more data points to train the model parameters on.

This reasoning might also apply for the varying number of recordings for the different cameras, while the main underlying difference is definitely their covered spectral range and their spatial resolution.



(a) *DeepHS Debris*

(b) HRSS

Fig. 5. Pixel-/patch-wise classification accuracy for (a) the *DeepHS Debris* (relevant classes) and (b) HRSS datasets (Indian Pines, Salinas, Pavia University - averaged) and each model considering the spatial dimension only (grey), the spectral dimension only (gold), or both dimensions (red) (Color figure online)

Overall, we conclude that models specialized on 3-dimensional HS data outperform regular vision models (like the ResNet), and that among those models, the larger ones are more sensitive to the size of the hyperspectral datasets and prone to overfitting. Smaller models, like the DeepHS-Nets, perform relatively better on the object-wise task, but are again outperformed by the more complex models when provided enough training samples. The two best performing models, the 3D CNN and HiT, are in fact also among the largest ones. The third-best model, DeepHS-Net+HyveConv, however is the smallest of all considered networks, with only 17.000 parameters.

Aside from their model type and size, the classifiers can be categorized based on which dimension(s) of the HSI cube they operate on. Figure 5 again shows the pixel- or patch-wise classification accuracy for the models and their dimensionality indicated by color, on our *DeepHS Debris* data subset as well as HRSS data for comparison.

We observe that pixel-based models that only access the spectral information, and purely spatial models, both achieve only low classification accuracies, while models like the DeepHS-Net variants or 3D CNN, operating on both, the spectral and spatial dimension, perform best overall. For the remote sensing data (Fig. 5(b)), we obtain an even more extreme distribution. We find that, here, the purely spatial information is already enough to achieve high accuracy (above 95%). This should be alarming to any researcher working in the field, as it indicates that the so frequently used HRSS dataset is not best suited for evaluating spectrum-based methods after all. Therefore, with our work, we aim to provide an alternative dataset and classification task(s), for which we show that, as expected, models considering both the spectral and spatial information can actually outperform purely spatial models for hyperspectral image classification (see Fig. 5(a)).

Also generally, model performance differs significantly for our and other hyperspectral datasets and applications. Especially, the ranking of methods changes for this dataset relative to existing ones, like for HRSS (see Fig. 5), but also for the *DeepHS Fruit v2* dataset.

Nonetheless, we can show that pre-training on our dataset is helpful for other HSI applications and tasks. Table 5, for example, lists the classification accuracy without and with pre-training on *DeepHS Debris* (Specim FX10 sensor), fine-tuned and evaluated on another sensor, another dataset, and even an entirely

Table 5. Classification accuracy for the DeepHS-Net+HyveConv, without pre-training (random initialization) and with pre-training on the *DeepHS Debris* dataset (Specim FX10/patch-wise), plus subsequent fine-tuning and evaluation on (a) a different sensor (*DeepHS Debris*/Corning HSI/patch-wise), (b) another dataset (*DeepHS Fruit v2*/Avocado/Corning HSI/object-wise), (c) a completely different task (HRSS/Indian Pines/0.05 train ratio/patch-wise). Highest accuracy in bold, respectively

	(a) Debris/Corning HSI	(b) Fruit	(c) HRSS
No pretraining	71.39%	88.89%	81.69%
Pre-trained on Debris/Specim FX10	**87.50**%	**100.00**%	**83.84**%

different application and task. Pre-training led to perfect classification on the *DeepHS Fruit v2* dataset and increased accuracy in all cases.

6 Limitations

While we present the first publicly available hyperspectral dataset of this specific kind, its size is still very limited compared to RGB datasets that usually contain thousands of images. Moreover, although it already covers a large majority of what can frequently be found on the construction site, the dataset could always be further expanded in terms of additional classes as well as additional samples for existing classes, esp. regarding intra-class variety.

There are critical cases in which a coating hides the actual material underneath (esp. glaze for tiles and ceramic). Also, samples in the dataset are not unmixed, but may contain other materials (like concrete which is made from cement and gravel or sand, or bricks partially covered by mortar), and therefore strictly speaking, cannot be clearly assigned to a single class.

This may cause problems, especially for purely spectral classification and pixel-wise prediction, as it was done by Serranti et al. [3,25]. In contrast, we pursue a different approach, namely to classify entire objects as one material type even if part of it is, e.g., partially covered by a different material. This focus on whole objects along with the increased number of materials makes classification on our dataset more difficult, but realistic and interesting for benchmarking at the same time.

7 Conclusion

We present a novel, publicly available hyperspectral dataset containing samples of 14 different construction waste material classes. Aside from serving as a benchmark for evaluating and comparing HSI classification approaches for both object-wise and patch- or pixel-wise prediction, our dataset might be used for pre-training HSI classifier networks for other, similar tasks.

Application-wise, this work can enhance automatic waste sorting and therefore the whole recycling process. Increasing the reusability of construction and demolition waste material represents a significant stride toward safeguarding our future environment.

Acknowledgements. This work has been supported by the German Ministry of Economy, Labour and Tourism, Project *KI-PRO-BAU*, FKZ: BW1_2108/02 in cooperation with Optocycle GmbH, Tuebingen. We would also like to thank Manuel Graña for the permission to use the *Hyperspectral Remote Sensing Scenes* (HRSS) dataset collection [9] for our experiments.

Data Availibility Statement. The *DeepHS Debris* dataset is publicly available for download via https://cogsys.cs.uni-tuebingen.de/webprojects/DeepHS-Debris-2024-Datasets/DeepHS-Debris-2024-Datasets.zip.

References

1. Abdulsamad, T., Chen, F., Xue, Y., Wang, Y., Yang, L., Zeng, D.: Hyperspectral image classification based on spectral and spatial information using resnet with channel attention. Opt. Quantum Electron. **53** (2021). https://doi.org/10.1007/s11082-020-02671-4
2. Ahmad, M., et al.: Hyperspectral image classification - traditional to deep models: a survey for future prospects. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. **15**, 968–999 (2022). https://doi.org/10.1109/JSTARS.2021.3133021
3. Bonifazi, G., Capobianco, G., Serranti, S., Palmieri, R.: Hyperspectral imaging applied to the waste recycling sector. Spectrosc. Eur. **31**, 8–11 (2019). https://doi.org/10.1255/sew.2019.a3
4. Bonifazi, G., Gasbarrone, R., Capobianco, G., Serranti, S.: A dataset of visible–short wave infrared reflectance spectra collected on pre-cooked pasta products. Data Brief **36**, 106989 (2021). https://doi.org/10.1016/j.dib.2021.106989
5. Chakrabarti, A., Zickler, T.: Statistics of real-world hyperspectral images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 193–200 (2011)
6. Chakraborty, T., Trehan, U.: SpectralNET: exploring spatial-spectral waveletCNN for hyperspectral image classification. arXiv abs/2104.00341 (2021)
7. Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y.: Deep learning-based classification of hyperspectral data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7**(6), 2094–2107 (2014). https://doi.org/10.1109/JSTARS.2014.2329330
8. Ghamisi, P., et al.: New frontiers in spectral-spatial hyperspectral image classification: the latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning. IEEE Geosci. Remote Sens. Mag. **6**(3), 10–43 (2018). https://doi.org/10.1109/MGRS.2018.2854840
9. Graña, M., Veganzons, M.A., Ayerdi, B.: Hyperspectral remote sensing scenes (2011). https://ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
10. Habili, N., et al.: A hyperspectral and RGB dataset for building facade segmentation (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, pp. 770–778. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.90
12. Hong, D., et al.: SpectralFormer: rethinking hyperspectral image classification with transformers. IEEE Trans. Geosci. Remote Sens. **60**, 1–15 (2022). https://doi.org/10.1109/TGRS.2021.3130716
13. Kalman, L.S., Bassett, E.M., III.: classification and material identification in an urban environment using hydice hyperspectral data. In: Descour, M.R., Shen, S.S. (eds.) Imaging Spectrometry III. SPIE (1997). https://doi.org/10.1117/12.283843
14. Khoshboresh-Masouleh, M., Hasanlou, M.: Improving hyperspectral sub-pixel target detection in multiple target signatures using a revised replacement signal model. Eur. J. Remote Sens. **53**, 316–330 (2020). https://doi.org/10.1080/22797254.2020.1850179
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015 (2015)

16. LeBauer, D., et al.: Data from: terra-ref, an open reference data set from high resolution genomics, phenomics, and imaging sensors (2020). https://doi.org/10.5061/DRYAD.4B8GTHT99
17. Lee, H., Eum, S., Kwon, H.: Exploring cross-domain pretrained model for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **60**, 1–12 (2022). https://doi.org/10.1109/TGRS.2022.3165441
18. Leone, G., Catarino, A., De Keukelaere, L., Bossaer, M., Knaeps, E., Everaert, G.: Flanders Marine Institute (VLIZ), Belgium, Flemish Institute For Technological Research (VITO), Belgium: Hyperspectral reflectance dataset for dry, wet and submerged plastics in clear and turbid water (2021). https://doi.org/10.14284/530
19. Lu, B., Dao, P., Liu, J., He, Y., Shang, J.: Recent advances of hyperspectral imaging technology and applications in agriculture. Remote Sens. **12**(16), 2659 (2020). https://doi.org/10.3390/rs12162659
20. Lu, G., Fei, B.: Medical hyperspectral imaging: a review. J. Biomed. Opt. **19**(1), 010901 (2014). https://doi.org/10.1117/1.jbo.19.1.010901
21. Md. Mansoor Roomi, S., Sathya Bama, B., Puvi Lakshmi, V., Vaishnavi, M.: Hyperspectral dataset of pure and pesticide-coated apples for measuring the level of fertilizers used. Data Brief **49**, 109321 (2023). https://doi.org/10.1016/j.dib.2023.109321
22. NCALM: 2013 IEEE GRSS Data Fusion Contest
23. Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A.J.: Deep learning classifiers for hyperspectral imaging: a review. ISPRS J. Photogramm. Remote. Sens. **158**, 279–317 (2019)
24. Roy, S.K., Krishna, G., Dubey, S.R., Chaudhuri, B.B.: HybridSN: exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **17**(2), 277–281 (2020). https://doi.org/10.1109/LGRS.2019.2918719
25. Serranti, S., Palmieri, R., Bonifazi, G.: Hyperspectral imaging applied to demolition waste recycling: innovative approach for product quality control. J. Electron. Imaging **24**(4), 043003 (2015). https://doi.org/10.1117/1.JEI.24.4.043003
26. Studier-Fischer, A., et al.: HeiPorSPECTRAL - the Heidelberg porcine HyperSPECTRAL imaging dataset of 20 physiological organs. Sci. Data **10**(1) (2023). https://doi.org/10.1038/s41597-023-02315-8
27. Varga, L.A., Frank, H., Zell, A.: Self-supervised pretraining for hyperspectral classification of fruit ripeness. In: 6th International Conference on Optical Characterization of Materials, OCM 2023, pp. 97–108. KIT Scientific Publishing (2023)
28. Varga, L.A., Makowski, J., Zell, A.: Measuring the ripeness of fruit with hyperspectral imaging and deep learning. In: International Joint Conference on Neural Networks, IJCNN 2021, pp. 1–8. IEEE (2021). https://doi.org/10.1109/IJCNN52387.2021.9533728
29. Varga, L.A., Messmer, M., Benbarka, N., Zell, A.: Wavelength-aware 2D convolutions for hyperspectral imaging. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, pp. 3777–3786. IEEE (2023). https://doi.org/10.1109/WACV56688.2023.00378
30. Wang, C., et al.: A review of deep learning used in the hyperspectral image analysis for agriculture. Artif. Intell. Rev. **54**(7), 5205–5253 (2021). https://doi.org/10.1007/s10462-021-10018-y
31. Waske, B., van der Linden, S., Benediktsson, J.A., Rabe, A., Hostert, P.: Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. **48**(7), 2880–2889 (2010). https://doi.org/10.1109/TGRS.2010.2041784

32. Windrim, L., Melkumyan, A., Murphy, R.J., Chlingaryan, A., Ramakrishnan, R.: Pretraining for hyperspectral convolutional neural network classification. IEEE Trans. Geosci. Remote Sens. **56**(5), 2798–2810 (2018). https://doi.org/10.1109/TGRS.2017.2783886
33. Yang, X., Cao, W., Lu, Y., Zhou, Y.: Hyperspectral image transformer classification networks. IEEE Trans. Geosci. Remote Sens. **60**, 1–15 (2022). https://doi.org/10.1109/TGRS.2022.3171551
34. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S.: Generalized Assorted Pixel Camera: Post-Capture Control of Resolution. Columbia University, Dynamic Range and Spectrum. Technical report (2008)

LiDUT-Depth: A Lightweight Self-supervised Depth Estimation Model Featuring Dynamic Upsampling and Triplet Loss Optimization

Hao Jiang[1], Zhijun Fang[1,2](✉), Xuan Shao[1](✉), Xiaoyan Jiang[2], and Jenq-Neng Hwang[3]

[1] School of Computer Science and Technology, Donghua University, Shanghai, China
zjfang@dhu.edu.cn

[2] School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China

[3] Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA

Abstract. Lightweight yet reliable depth estimation models that can deployed on edge devices are crucial for the practical application of fields such as autonomous driving, robot navigation, and augmented reality. However, previous research often fails to balance accuracy and complexity well. Existing lightweight models still have relatively high error rates in specific scenarios, which makes them unsuitable for industrial applications. Hence, we propose LiDUT-Depth, a lightweight architecture for self-supervised monocular depth estimation that is enhanced through dynamic upsampling and triplet loss optimization, which achieves competitive results with smaller model sizes and lower computational complexity. Specifically, an efficient dynamic upsampling module (EDU Module) is employed to preserve detailed local features, thereby obtaining more accurate depth estimation results. In addition, an improved semantic-aware triplet loss (SaTri Loss) is integrated into the training loss, significantly improving depth estimation accuracy in gradient-rich regions. Experiments show that our architecture achieves a considerably enhanced accuracy compared with previous works with a much lower model size. Our codes and models are available here.

Keywords: Self-supervised · Monocular depth estimation · Lightweight architecture · Efficient dynamic upsample · Semantic-aware triplet loss

This work was supported in part by the Natural Science Foundation of China under Grant U2033218, in part by the Shanghai Sailing Program under Grant 23YF1400900, and in part by the Fundamental Research Funds for the Central Universities under Grants CF221223219597.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 176–189, 2025.
https://doi.org/10.1007/978-3-031-78444-6_12

1 Introduction

Pixel-wise dense depth maps are widely used in fields such as autonomous driving, robot navigation, and augmented reality. However, LiDAR-based depth data is expensive to acquire, leading to research on using deep neural networks to estimate dense depth maps from an RGB image [6,7,13]. Traditional supervised methods require extensive, accurate ground truth depth data, which is time-consuming and costly. This has led to increased interest in self-supervised learning methods, which transform depth estimation into a view synthesis problem using neighboring views or frames as supervision signals [10,11,21,24]. Thus, In this paper, we focus on self-supervised training using monocular videos. Unlike existing works that are eager to build deeper and more complex CNN architectures [21,25], our approach prioritizes practical applications, i.e., we focus not only on accuracy but also on light-weighting. Given the aforementioned considerations, we propose LiDUT-Depth, a lightweight yet reliable self-supervised monocular depth estimation architecture. Our architecture employs an efficient dynamic upsampling module (EDU Module) to preserve detailed features. In addition, to optimize our model and improve the estimation accuracy in gradient-rich areas, we also incorporate an improved semantic-aware triplet loss (SaTri Loss) into the loss calculation. Our architecture builds upon the baseline work [22], enhancing estimation reliability without adding any computational overhead at inferring time.

The main contributions of this work can be summarized in three aspects:

- A lightweight architecture for self-supervised monocular depth estimation, named LiDUT-Depth, is proposed, further improving depth estimation accuracy without adding computational overhead or increasing model size.
- An efficient dynamic upsampling module (EDU Module) is introduced to preserve more detailed local features, thereby obtaining more accurate depth estimation results.
- An improved semantic-aware triplet loss (SaTri Loss) is integrated into the training loss to optimize the model further, significantly improving the model's performance in gradient-rich regions without affecting the inference speed.

2 Related Work

In this section, we reviewed the tasks of monocular depth estimation using deep learning and dense feature extraction methods separately.

2.1 Monocular Depth Estimation Using Deep Learning

Monocular depth estimation is a fuzzy and ill-posed problem, as infinite world scenes can generate a given image. Deep learning methods for depth estimation can be broadly categorized into supervised and self-supervised learning.

Supervised Depth Estimation. Supervised depth estimation treats the task as regression, using ground truth depth maps as supervision signals. It utilizes deep neural networks to extract features from input images and learns the relationship between depth and RGB values. Eigen et al. [6] proposed a multi-scale network stack that combines features from global coarse-scale and local fine-scale networks, introducing scale-invariant error and achieving depth estimation from a single image using deep neural networks (DNNs). Laina et al. [13] introduced the inverse Huber loss for model optimization (Fig. 1).

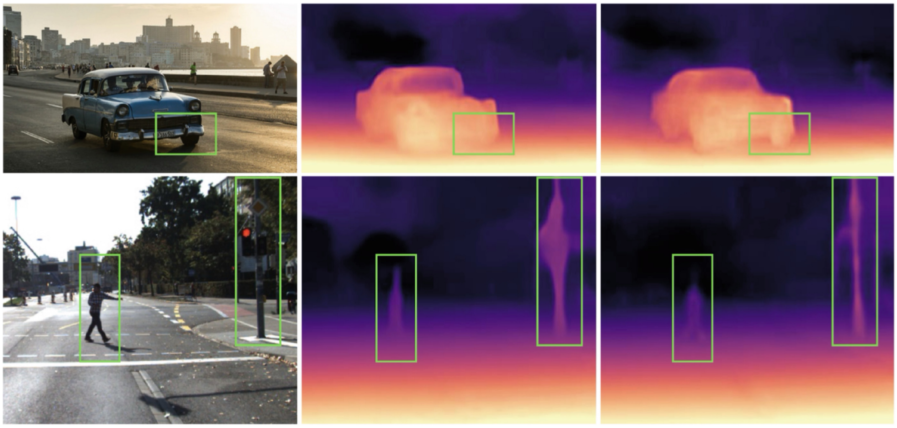


Fig. 1. Depth estimation comparisons. From left to right are the input image and the depth maps predicted by Lite-Mono [22], and LiDUT-Depth (ours), respectively. Our architecture shows higher accuracy, especially in gradient-rich areas

Self-supervised Depth Estimation. Self-supervised depth estimation methods were initially proposed by Garg et al. [8], who treated depth estimation from stereo images as a view synthesis problem. They enabled depth estimation without ground truth data by minimizing the photometric reprojection loss between the input left view and the output right view. Zhou et al. [24] introduced the SfM-Learner architecture, which incorporated a camera pose estimation network to estimate depth from monocular video frames. Godard et al. [11] proposed the Monodepth2 architecture, which addressed occlusions and motion by optimizing the loss function and introducing masking, achieving high accuracy without increasing network parameters. Additional supervision techniques have been introduced, including optical flow estimation [21] and spatial-temporal geometric constraints [19]. In recent years, Vision Transformers (ViTs) have shown remarkable performance in computer vision tasks, leading to their integration into depth estimation. Zhao et al. [23] proposed an architecture that combines CNN with Transformers, resulting in more detailed and accurate predictions. Zhang et al. [22] introduced Continuous Dilated Convolution (CDC) modules and Local-Global Feature Interaction (LGFI) modules, reducing trainable parameters while improving accuracy.

2.2 Feature Upsampling

In the widely used encoder-decoder architecture of depth estimation, the decoder typically needs to receive the feature maps extracted by the encoder, then decode them separately at 3–5 different scales, upsample and concatenate them, and feed them to the next layer.

The two most commonly used upsampling methods, NN (Nearest Neighbor) and Bilinear Interpolation, both ignore the semantic meaning in the feature maps and only use fixed rules to interpolate low-resolution features [14], which will result in the loss of a large number of valuable features during the depth decoding process. Researchers have made various attempts to overcome these issues. SegNet [1] uses max pooling to retain more edge information, but the introduced zero padding disrupts the semantic continuity of smooth areas. Pixel Shuffle [18] first uses convolution to increase the number of channels and then reshapes the feature maps to improve the resolution, but this inadvertently increases the model's size, making it more difficult to deploy on edge devices.

Therefore, in order to retain more details of the feature maps at each scale and thus estimate more accurate depth maps, our goal is to replace the commonly used bilinear interpolation upsampling module in the depth decoder with a more efficient upsampling method.

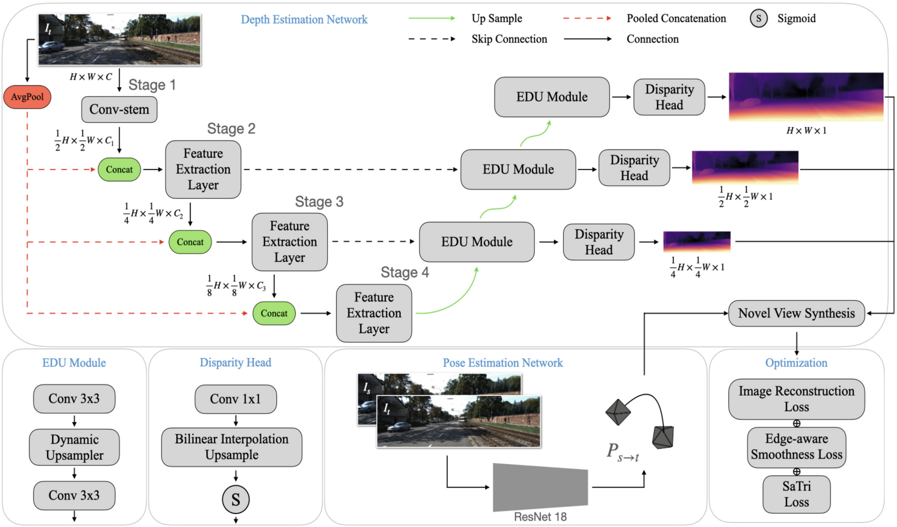


Fig. 2. Overview of the proposed LiDUT-Depth. LiDUT-Depth consists of an encoder-decoder depth estimation network and a pose estimation network similar to [11]. The decoder of the depth estimation network employs an EDU Module to preserve more detailed local features. The model was also optimized using an additional SaTri Loss

3 Methodology

As Fig. 2 shows, LiDUT-Depth consists of two networks: a depth estimation network with an encoder-decoder structure for estimating the depth map at different scales from a single input RGB image, and a pose estimation network for estimating the relative camera pose between adjacent frames. The information obtained from the two networks is used to reconstruct the target view, thus calculating the error with the actual target view and optimizing the model. Then, in the following sections, we demonstrated the aforementioned two networks in detail, and provided a thorough explanation of the SaTri Loss we integrated into our training loss.

3.1 Depth Estimation Network

Similar to most prior work, we employ an encoder-decoder U-Net architecture to design our Depth Estimation Network.

Depth Encoder. We use a depth encoder adopted from Lite-Mono [22]. Using a 4-stage hybrid architecture of CNN and Transformer, the encoder can extract rich, detailed features while also encoding long-range global information into the features.

Depth Decoder. Considering that the U-Net architecture requires multiple downsampling and upsampling operations, which may lead to significant detail loss, it is crucial to minimize this loss as much as possible. Therefore, [22] introduced dilated convolutions and Transformers in the encoder to obtain a larger receptive field, addressing this issue. However, in the decoder, they only used simple bilinear interpolation for upsampling, resulting in substantial information loss in the feature maps. To address this, we employed a more efficient dynamic upsampling module (EDU Module) as the upsampler. This approach mitigates feature loss without requiring additional attention mechanisms [23] or complex upsampling methods [25].

EDU Module. The EDU Module is introduced to retain more detailed local features in the decoding phase. Inspired by image super-resolution methods, we employ the low computational cost and highly effective Pixel Shuffle [18] as the core upsampling algorithm in the EDU Module. Specifically, given a feature map $\mathcal{X}$ with size $C \times H \times W$ and an upsampling ratio s (which is 2 in our architecture), a feature map $\mathcal{X}'$ with size $C \times sH \times sW$ is to be generated. We use two linear layers with both input channels of C and output channels of $2s^2$ to compute the offsets. First, we use the first linear layer, $Linear_a$, to calculate an initial offset $\mathcal{O}$ with size $2s^2 \times H \times W$. Second, to enhance the flexibility of this initial offset, we employ a second linear projection layer, $Linear_b$, to obtain a dynamic weight of size $2s^2 \times H \times W$, and apply a Hadamard product with the offset $\mathcal{O}$ to achieve a new dynamic offset $\mathcal{O}$. Then, we reshape $\mathcal{O}$ to $2 \times sH \times sW$ through Pixel Shuffle. Finally, by calculating the sum of the offset $\mathcal{O}$ and the original sampling grid $\mathcal{G}$, we obtain a sampling set s, as shown in Fig. 3.

$$\mathcal{O} = 0.5 \times Sigmoid(Linear_a(\mathcal{X})) \cdot Linear_b(\mathcal{X}), \tag{1}$$

$$\mathcal{S} = \mathcal{O} + \mathcal{G}. \tag{2}$$

In the end, the $\mathcal{GS}$ function, which stands for the built-in grid-sample function of PyTorch, takes the positions in $\mathcal{S}$ and uses them to resample the hypothetical bilinear-interpolated feature map $\mathcal{X}$ into $\mathcal{X}'$:

$$\mathcal{X}' = \mathcal{GS}(\mathcal{X}, \mathcal{S}). \tag{3}$$

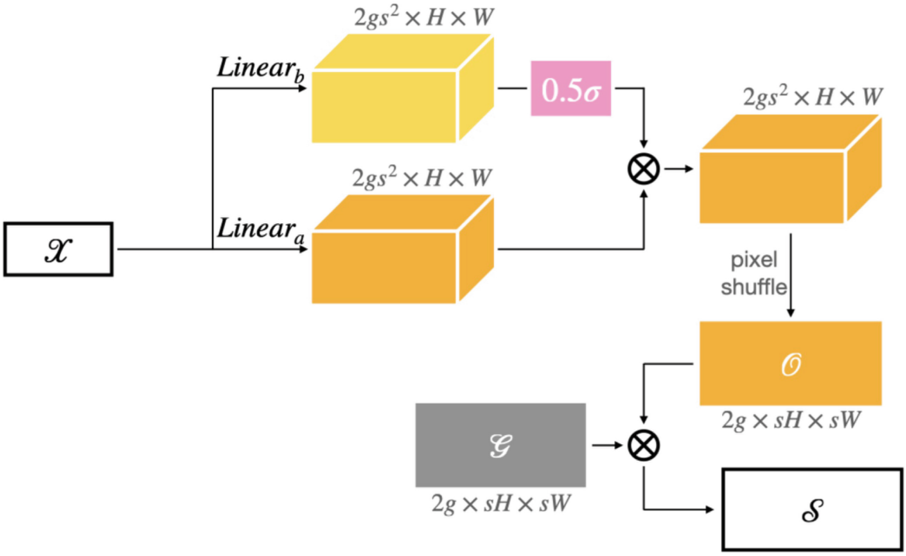


Fig. 3. Sampling set generator in the EDU module. The EDU module uses the positions in the sampling set $\mathcal{S}$ generated here to upsample the input feature maps

3.2 Pose Estimation Network

For the sake of lightweight design, our work follows the approach used in [11], using an ImageNet pre-trained ResNet18 as the pose encoder to encode the image pairs of adjacent frames in the video sequence as input. Then, we use a pose decoder with four convolutional layers to estimate the 6-DoF relative camera pose between two frames.

3.3 Self-supervised Learning

Following other self-supervised learning methods, we transform the depth estimation task into a novel view synthesis task. Specifically, the learning objective is to minimize an image reconstruction loss $\mathcal{L}_r$ between a target view I_t and the synthesized target view I'_t, together with an edge-aware smoothness loss $\mathcal{L}_s$.

Moreover, we also introduced $\mathcal{L}_{tri}$, an improved semantic-aware triplet loss [3] (SaTri Loss), to optimize the model, thereby improving the model's performance in gradient-rich regions.

Image Reconstruction Loss. First, we take three frames from a sequence of consecutive video frames. The middle frame is denoted as the target view I_t, and either of the adjacent frames can be considered source view I_s. Next, we define D_t as the predicted depth map, $P_{s\rightarrow t}$ as the predicted relative camera pose between the two frames, and K as the camera intrinsic parameters. Then, similar to [11], we denote the image reconstruction loss as:

$$\mathcal{L}_r(I'_t, I_t) = \mu \cdot \mathcal{L}_p(I'_t, I_t), \tag{4}$$

where I_t is the target view, I_s is the adjacent source view, I'_t is the reconstructed target view, $\mathcal{L}_p$ stands for the photometric reprojection loss, and $\mu = \min \mathcal{L}_p(I_s, I_t) < \min \mathcal{L}_p(I'_t, I_t)$ represents the binary masks used to remove moving pixels.

Edge-Aware Smoothness Loss. Following [11,25], we calculate $\mathcal{L}_s$ as the edge-aware smoothness loss to obtain a smoother depth map:

$$\mathcal{L}_s = |\partial_x d^*_t| e^{-|\partial_x I_t|} + |\partial_x d^*_t| e^{-|\partial_y I_t|}, \tag{5}$$

where $d^*_t = \frac{d_t}{d'_t}$ represents the inverse depth normalized by the mean.

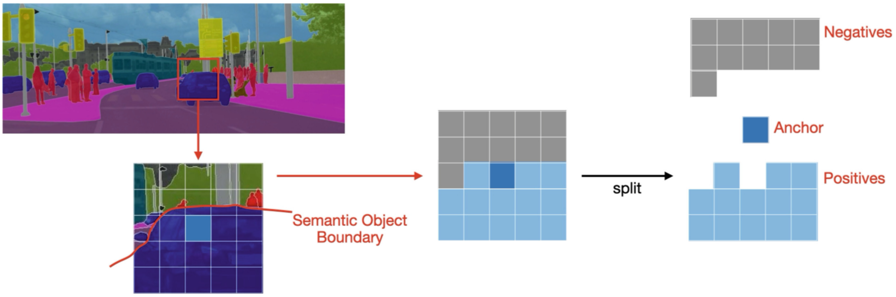


Fig. 4. Splitting patches according to semantic information. For pixels in the semantic boundary region, we split the local patch into a triplet according to the semantic patch

SaTri Loss. In addition, we noticed that the depth information in gradient-rich areas, namely the edge regions, is often prone to prediction errors, as is called edge-fattening. Therefore, we introduce the SaTri Loss to optimize this issue without increasing the additional computational overhead during the inference stage [3]. However, it should be noted that when we initially introduced this loss to the baseline work [22], the experimental results demonstrated that its optimization effect on the model was extremely limited, the *AbsRel* metric only decreases from 0.107 to 0.106. We believe this is due to the use of bilinear interpolation for upsampling in the network, which results in the loss of a significant amount of detail in the feature maps, particularly in the edge regions of objects. This negatively impacts the subsequent optimization of the model when combining semantic information. However, after we optimized the network architecture to address this issue by incorporating the EDU Module with Pixel Shuffle in the DepthNet decoder, the optimization effect of this loss was significantly enhanced. Subsequent ablation experiments (Table 3) also confirmed this improvement.

As shown in Fig. 4, we first group the pixels in a local patch into triplets, defining the center pixel as the anchor point $\mathcal{P}_i$, pixels with the same semantic meaning as the positive sample $\mathcal{P}_i^+$, and others as the negative sample $\mathcal{P}_i^-$. Then, define the anchor-positive sample distance $\mathcal{D}^+$ and the anchor-negative sample distance $\mathcal{D}^-$ as the mean of the Euclidean distance of L_2 normalized deep features [12]:

$$\mathcal{D}^+(i) = \frac{1}{|\mathcal{P}_i^+|} \sum_{j \in \mathcal{P}_i^+} \|\hat{F}_d(i) - \hat{F}_d(j)\|_2^2, \tag{6}$$

$$\mathcal{D}^-(i) = \frac{1}{|\mathcal{P}_i^-|} \sum_{j \in \mathcal{P}_i^-} \|\hat{F}_d(i) - \hat{F}_d(j)\|_2^2, \tag{7}$$

where $\hat{F}_d = \frac{F_d}{\|F_d\|}$, in which F_d represents the corresponding depth feature. Then, the loss function will minimize the anchor-positive distance $\mathcal{D}^+$ and maximize the anchor-negative distance $\mathcal{D}^-$. Furthermore, a margin m is introduced as a threshold to regulate the minimum separation between D^- and D^+, thereby preventing an excessive discrepancy between the two:

$$\mathcal{D}^- - \mathcal{D}^+ > m, \tag{8}$$

Then, the original triplet loss is defined as:

$$\mathcal{L}_{tri} = \frac{1}{|\gamma|} \sum_{\mathcal{P}_i \in \gamma} [D^+(i) - \mathcal{D}^-(i) + m]_+, \tag{9}$$

where $[\cdot]_+$ is the hinge function.

However, the original triplet loss suffers from two issues: it may overlook small but poorly estimated fatten regions, and there also exists a mutual influence between positive and negative samples. To address these problems, following the work of Chen et al. [3], we apply a strategy based on the minimal operator to handle all negative samples. This prevents well-performing negative samples from masking errors from margin-inflating negative samples. Additionally, we separate the anchor-positive distance and anchor-negative distance from the original triplet and directly optimize the positive samples, avoiding the influence of negative samples. By doing so, we obtain an improved triplet loss called SaTri Loss:

$$\mathcal{L}_{satri} = \frac{1}{|\gamma|} \sum_{\mathcal{P}_i \in \gamma} \left(\mathcal{D}^+(i) + [m' - \mathcal{D}^{-'}(i)]_+ \right), \tag{10}$$

where $\gamma = \mathcal{P}_i | (|\mathcal{P}_i^+| > k) \wedge (|\mathcal{P}_i^-| > k)$ is the set that contains all the semantic boundary pixels that satisfy the aforementioned constraint conditions.

Total Loss. Finally, we obtain this complete loss function:

$$\mathcal{L} = \frac{1}{3} \sum_{s \in [1, \frac{1}{2}, \frac{1}{4}]} (\mathcal{L}_r + \lambda \mathcal{L}_s + \mathcal{L}_{satri}), \tag{11}$$

where s represents different scales outputted by the decoder, with λ set to $1e^{-3}$, which is the same value as [11].

4 Experiments

4.1 Dataset

The KITTI [9] dataset contains 61 road scenes and collects a large amount of data using multiple sensors, including RGB cameras, LiDAR, GPU/IMU, etc. We used the Eigen split [5] for training and evaluation. In the monocular frame sequence, every three consecutive frames were taken as a group, with 39,180 triplets used for training, 4,424 for evaluation, and 697 for testing. Self-supervised training is based on the known camera intrinsic matrix. Following the approach of Godard et al. [11], we took the average focal length of all images in the KITTI dataset, thus using the same intrinsic parameter value for all images during training. In line with common practice, in the evaluation, we limit the maximum depth of the forecast to 100 m and the minimum to 0 m.

4.2 Implementation Details

Our model was implemented using PyTorch, with AdamW [15] as the optimizer. We loaded the weights pre-trained on ImageNet [4] and trained the model for 22 epochs. The batch size was set to 12, the initial learning rate was set to $1e^{-4}$, and the weight decay was set to $1e^{-2}$. The input/output resolution was set to 640×192. On a single RTX 3080 GPU, the training time for 22 epochs is approximately 13 h.

4.3 Quantitative and Qualitative Results on KITTI

As shown in Table 1, we have quantitatively compared our architecture with other representative methods, and the qualitative depth estimation results compared with former state-of-the-art methods are also illustrated in Fig. 5. LiDUT-Depth demonstrates the best balance between accuracy and complexity among all methods. Compared to the larger version of Monodepth2 [11], which uses ResNet-50 as the backbone network, our architecture achieves a 6.36% error reduction in terms of AbsRel in 640×192 resolution, with less than one-tenth of its size. Our architecture also outperforms the previous lightweight model R-MSFM [25]. Compared to Lite-Mono [22], we reduced AbsRel by 3.63% without increasing Params or FLOPs.

Table 1. Comparison of LiDUT-Depth to prior competitors on KITTI using the Eigen split [5]. All input images are resized to 640 × 192. The 1^{st} and the 2^{nd} best results are highlighted in **bold** and underlined, respectively

Method	Year	Data	Depth Error (↓)				Depth Accuracy (↑)			Model Size (↓)
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	Params.
GeoNet [21]	2018	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975	31.6M
Monodepth2 ResNet-18 [11]	2019	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981	14.3M
Monodepth2 ResNet-50 [11]	2019	M	0.110	0.831	4.642	0.187	0.883	0.962	0.982	32.5M
HR-Depth [16]	2021	M	0.109	0.792	4.632	0.185	0.884	0.962	0.983	14.7M
R-MSFM3 [25]	2021	M	0.114	0.815	4.712	0.193	0.876	0.959	0.981	3.5M
R-MSFM6 [25]	2021	M	0.112	0.806	4.704	0.191	0.878	0.960	0.981	3.8M
MonoFormer [2]	2021	M	0.108	0.806	4.594	0.184	0.884	0.963	0.983	>23.9M
Lite-Mono [22]	2023	M	0.107	0.765	4.561	0.183	0.886	0.963	0.983	3.1M
LiDUT-Depth (Ours)	2023	M	**0.103**	**0.723**	**4.469**	**0.178**	**0.889**	**0.964**	**0.984**	**3.1M**

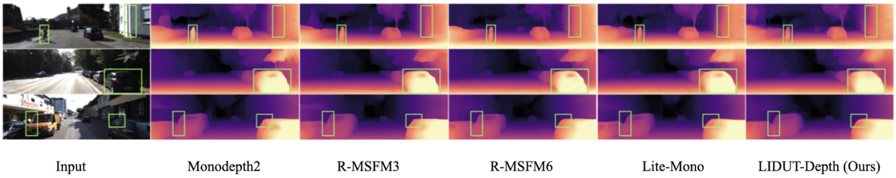


Fig. 5. Qualitative results on KITTI. From left to right are the input image and the depth maps predicted by Monodepth2 [11], R-MSFM3 [25], R-MSFM6 [25], Lite-Mono [22], and LiDUT-Depth (ours), respectively. Other methods have limited accuracy in estimating edge area, whereas our model can achieve better results

4.4 Quantitative and Qualitative Results on Make3D

Zero-shot experiments are also conducted on the Make3D dataset to verify the generalization ability of the proposed method in different outdoor scenes. As shown in Table 2, LiDUT-Depth outperforms the other four methods. Figure 6 shows the visual comparison results. Thanks to the proposed EDU module and SaTri Loss, LiDUT-Depth can achieve more accurate depth predictions in regions with large gradients.

4.5 Ablation Studies

To further demonstrate the effectiveness of the proposed architecture, we conducted an ablation analysis of the introduced components, and the results are shown in Table 3. It is imperative to acknowledge that the model size remains unaffected by the proposed method, thereby resulting in an equivalent number of model parameters across all ablation experiments.

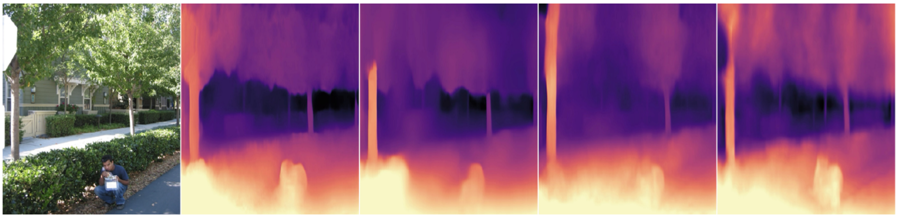


Fig. 6. Qualitative results on Make3D [17]. From left to right are the input image and the depth maps predicted by Monodepth2 [11], R-MSFM6 [25], Lite-Mono [22], and LiDUT-Depth (ours), respectively

Table 2. Comparison of LiDUT-Depth to prior competitors on the Make3D [17] **Dataset**. All models are trained on KITTI [9] with the resolution of 640 × 192

Method	Abs Rel	Sq Rel	RMSE	RMSE log
DDVO [20]	0.387	4.720	8.090	0.204
Monodepth2 [11]	0.322	3.589	7.417	0.163
R-MSFM6 [25]	0.334	3.285	7.212	0.169
Lite-Mono [22]	0.305	3.060	6.981	0.158
LiDUT-Depth (Ours)	**0.301**	**2.935**	**6.947**	**0.157**

Table 3. Ablation study on model architectures. All the models mentioned above are trained and tested on KITTI with the input size of 640 × 192

Architecture	Params.	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
LiDUT-Depth full model	3.069M	0.103	0.723	4.469	0.178	0.889	0.964	0.984
w/o EDU Module	3.069M	0.106	0.726	4.419	0.178	0.888	0.964	0.984
w/o SaTri Loss	3.069M	0.111	0.716	4.477	0.183	0.872	0.961	0.984

EDU Module. The accuracy suffers a 2.91% decrease in terms of Abs Rel when all the EDU Modules in the depth decoder are removed and replaced with the original bilinear interpolation upsampling. This is because bilinear interpolation upsampling results in the loss of a significant amount of detail in the feature maps, especially at object edges. However, these details are crucial for the SaTri Loss, which combines semantic information to compute the loss during the model optimization phase. Thus, the EDU module effectively helps the model retain more local features during the decoding stage, resulting in smoother feature maps at the edges and maximizing the effectiveness of subsequent model optimization.

SaTri Loss. When optimizing the model without the SaTri Loss, the accuracy drops significantly on all metrics except for Sq Rel. Specifically, the Abs Rel (lower is better) increased by 7.76%, even higher than the baseline model. Thus, the triplet loss computed in combination with semantic information can help the model more accurately estimate depth in gradient-rich regions.

It is important to note that the above experiments validate our optimization approach, that Simply combining semantic information to compute the loss for model optimization may not yield significant performance improvements because a substantial amount of useful information, including object edge features, is lost in the decoder, which makes it difficult to align with semantic boundaries during the optimization process. However, by optimizing the network architecture and employing a more effective upsampling method, we can retain this information, resulting in clearer edges and achieving the best optimization results.

5 Conclusion

This paper proposes a lightweight self-supervised monocular depth estimation framework named LiDUT-Depth. The depth network of this architecture features an EDU Module for preserving more detailed local features during upsampling. In addition, the architecture introduces a SaTri Loss to optimize the model, thereby obtaining depth estimation results with more precise edges. Experiments on the KITTI dataset demonstrate the superiority of our architecture in accuracy, as well as the excellent balance we achieve in model size and computational complexity.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
2. Bae, J., Moon, S., Im, S.: Deep digging into the generalization of self-supervised monocular depth estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 187–196 (2023)
3. Chen, X., Zhang, R., Jiang, J., Wang, Y., Li, G., Li, T.H.: Self-supervised monocular depth estimation: solving the edge-fattening problem. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5776–5786 (2023)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
5. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658 (2015)
6. 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, vol. 27 (2014)

7. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2002–2011 (2018)
8. 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
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
10. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 270–279 (2017)
11. 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)
12. Jung, H., Park, E., Yoo, S.: Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12642–12652 (2021)
13. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 239–248. IEEE (2016)
14. Liu, W., Lu, H., Fu, H., Cao, Z.: Learning to upsample by learning to sample. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6027–6037 (2023)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
16. Lyu, X., et al.: HR-depth: high resolution self-supervised monocular depth estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2294–2301 (2021)
17. Saxena, A., Sun, M., Ng, A.Y.: Make3D: learning 3D scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 824–840 (2008)
18. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874–1883 (2016)
19. Wang, A., et al.: Unsupervised learning of depth and ego-motion with spatial-temporal geometric constraints. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 1798–1803 (2019)
20. Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2022–2030 (2018)
21. Yin, Z., Shi, J.: Geonet: unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1983–1992 (2018)
22. Zhang, N., Nex, F., Vosselman, G., Kerle, N.: Lite-mono: a lightweight CNN and transformer architecture for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18537–18546 (2023)
23. Zhao, C., et al.: MonoViT: self-supervised monocular depth estimation with a vision transformer. In: 2022 International Conference on 3D Vision (3DV), pp. 668–678. IEEE (2022)

24. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1851–1858 (2017)
25. Zhou, Z., Fan, X., Shi, P., Xin, Y.: R-MSFM: recurrent multi-scale feature modulation for monocular depth estimating. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12777–12786 (2021)

Generalization Gap in Data Augmentation: Insights from Illumination

Jianqiang Xiao[1,2](✉), Weiwen Guo[3], Junfeng Liu[1], and Mengze Li[4]

[1] South China University of Technology, Guangzhou, China
xiaojianqiang0325@hotmail.com
[2] Hitachi Elevator (China) Co., Ltd., Guangzhou, China
[3] Hitachi Building Technology (Guangzhou) Co., Ltd., Guangzhou, China
[4] Research Center for Frontier Fundamental Studies, Zhejiang Lab, Hangzhou, China

Abstract. In the field of computer vision, data augmentation is widely used to enrich the feature complexity of training datasets with deep learning techniques. However, regarding the generalization capabilities of models, the difference in artificial features generated by data augmentation and natural visual features has not been fully revealed. This study introduces the concept of "visual representation variables" to define the possible visual variations in a task as a joint distribution of these variables. We focus on the visual representation variable "illumination", by simulating its distribution degradation and examining how data augmentation techniques enhance model performance on a classification task. Our goal is to investigate the differences in generalization between models trained with augmented data and those trained under real-world illumination conditions. Results indicate that after applying various data augmentation methods, model performance has significantly improved. Yet, a noticeable generalization gap still exists after utilizing various data augmentation methods, emphasizing the critical role of feature diversity in the training set for enhancing model generalization.

Keywords: Computer Vision · Data Augmentation · Generalization

1 Introduction

Over the past ten years, there has been a significant revolution in computer vision field. The advancement mainly belongs to deep learning techniques, particularly the utilization of Convolutional Neural Networks (CNNs) [1–4] and Transformer [5,6] architectures. By emulating creatural visual systems, CNNs stack convolutional layers and pooling layers, while Transformers utilize self-attention mechanisms [7] to handle long-range image dependencies effectively. With all these efforts, researchers made significant progress in image classification [1–4], object detection [8,9], and semantic segmentation [10,11]. Nowadays, computer vision expands its applications in industry [12], healthcare [13], and transportation [14], improving convenience and quality of people's lives.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 190–205, 2025.
https://doi.org/10.1007/978-3-031-78444-6_13

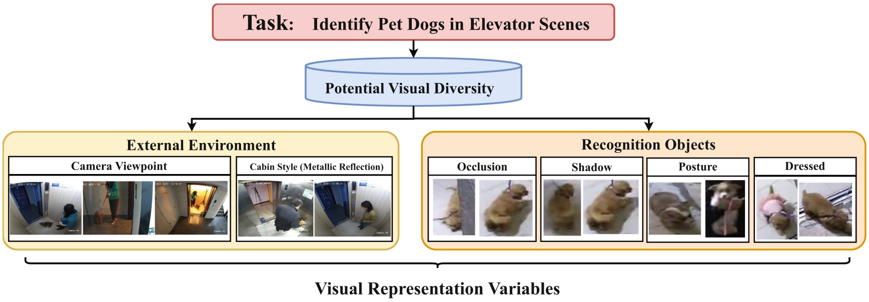

Task: Identify Pet Dogs in Elevator Scenes
Potential Visual Diversity
External Environment
Camera Viewpoint
Cabin Style (Metallic Reflection)
Recognition Objects
Occlusion
Shadow
Posture
Dressed
Visual Representation Variables

Fig. 1. Visual representation variables decomposition guided by task prior knowledge

In the realm of deep learning-based computer vision algorithms, the quality and variety of data significantly impact the generalization of visual models [15,16]. Despite the importance of real-world data collection, challenges such as scene diversity, labeling expenses, and privacy issues hinder the ability of extensive datasets to fully capture all visual characteristics. Through techniques such as geometric modifications, color channel adjustments, and filter incorporation [17,18], data augmentation effectively enhances visual feature diversity in datasets, improving training model generalization capabilities. Furthermore, data augmentation serves as a crucial method to prevent overfitting, especially when dealing with limited data. Consequently, data augmentation has become a fundamental component in the training process of computer vision application projects [19–21].

The visual diversity in datasets can be broadly divided into two categories: changes in the intrinsic properties of the recognition objects (e.g., the variety of vehicles in autonomous driving scenarios [22]), and changes indirectly caused by external environmental factors (e.g., different weather conditions [23] in self-driving). The visual diversity in the dataset ensures the model preserves robust recognition capabilities across various scenarios. However, when the distribution of visual characteristics is uneven, potentially diminishes the model's performance across various environmental conditions [24]. When data distribution imbalances occur, data augmentation is widely used as an effective method [25–27]. However, a discernible gap remains between augmented images and those captured in the physical world, highlighting under certain extreme conditions, such as adverse weather, the artificial features produced by data augmentation could potentially impair model performance [28,29]. This has prompted a reevaluation of data augmentation, whether synthetic, non-realistic pixel-wise feature characteristics might undermine a model's generalization in real-world scenario. Our study involves controlling illumination settings in a classification task to compare model performance under real-world and data augmentation datasets, aiming reveal the effectiveness and limitations of data augmentation.

In our study, we introduce the concept of "visual representation variables" to define the potential visual changes in a task as a joint distribution of these variables (Fig. 1). We focus on isolating a single visual representation variable, "illumination", to comprehensively study its effects. "Illumination" was chosen because it can be quantitatively measured and is relatively easy to replicate in data augmentation. By controlling illumination settings in a classification task, we compare model performance under real-world and data augmentation conditions to reveal the effectiveness and limitations of data augmentation.

Our main contributions are as follows:

- We validate that the model's generalization ability suffers a devastating impact when the illumination environment degrades into a singular distribution;
- By using a gray card to measure scene illumination mapping and optimizing color data augmentation parameters through Bayesian optimization, we achieve significant improvements in the generalization ability of datasets with singular illumination distribution;
- We demonstrate a significant generalization gap between models trained with data augmentation and those trained with real-world data. This emphasizes the limitations of data augmentation in replicating real-world visual features and underscores the necessity of carefully designed datasets.

2 Related Works

2.1 Domain Generalization

Domain generalization aims to enhance a model's performance on unseen data (target domain), particularly when there is a distributional discrepancy between the training data (source domain) and test data. The main strategies include domain alignment [30] to minimize distributional differences between source and target domains, meta-learning [31] which leverages learning across tasks to boost generalization capabilities, data augmentation [32,33] that introduces sample visual diversity to enhance model robustness and ensemble learning [34] which integrates multiple models to optimize overall performance. The issue of domain shift in domain generalization can be decomposed into differences in the distribution of a series of "visual representation variables" between the source domains and target domains. Our study focuses on data-level augmentation as it directly enriches the visual features of training data without the need to alter network structures or training strategies, closely aligning with our goal of exploring the impact of illumination environments on model generalization.

2.2 Data Augmentation

Data augmentation is a crucial technique for enhancing the generalization ability of deep learning models. Common methods, such as geometric transformations [35], color space adjustments [36], and random cropping [37], expand the representation of visual features and strengthen the model's generalization ability to recognize unseen data. Specifically, adjustments in the color space simulate different lighting conditions, directly influencing the model's adaptability to

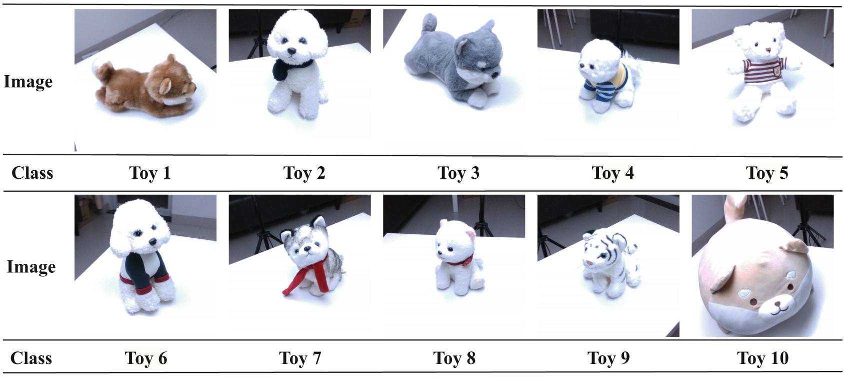


Fig. 2. An assortment of 10 distinct toy dogs serves as recognition targets in our classification task. The variety in their visual features, such as shape, color, fur texture, and attire, highlights the complexity of our dataset and assesses the classification models' ability to distinguish visual differences from subtle to pronounced

changes in illumination [38]. Although current research on data augmentation mainly focuses on its efficacy in enhancing model generalization, there has been limited exploration of the holistic impact of data augmentation techniques in simulating complex environmental changes, such as illumination settings [39]. Our study focuses on illumination settings, investigating the generalization effects of models by constructing distributions of real lighting and corresponding data-augmented distributions. This approach aims to reveal the potential disparities in generalization capabilities between augmented datasets and real-world datasets.

3 Experimental Framework and Data Preparation

3.1 Recognition Targets

To verify the universality of our insights, we selected basic image classification as our target task. As shown in Fig. 2, the collection of objects comprises ten different toys designed to challenge vision models' ability to recognize a wide range of visual aspects. The diversity in shape, color, and texture of these toys spans from easily recognizable to complex features, thoroughly testing the models' capabilities in processing subtle visual differences. This systematic selection strategy evaluates the models' recognition ability to handle visual complexity and reveals the delicate differences in visual features during the generalization process, enhancing our understanding of the image classification mechanism.

Given that this study primarily investigates the impact of data augmentation techniques on enhancing model generalization capabilities, focusing especially on the challenge of the "illumination" variable compared to real datasets, we aim to dive into how models adapt to visual stimuli across different data augmentation methodologies. By selecting toys with distinct visual characteristics as recognition objects, we established a foundation for subsequent research into data augmentation and model generalization.

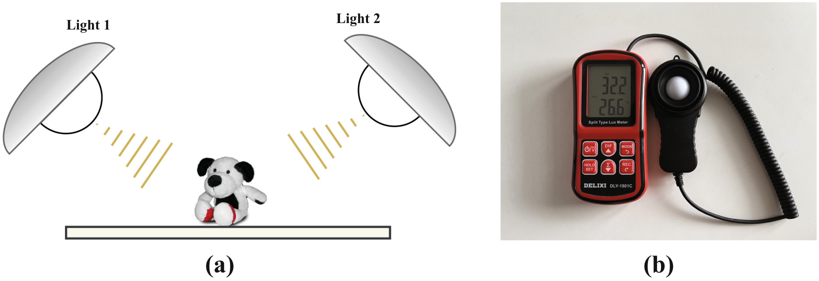


Fig. 3. (a) A dual light source setup with supplementary lamps placed at 45-degree angles to ensure balanced illumination. (b) The light intensity meter for precise measurement of illumination conditions

3.2 Illumination Environment Setting

To examine the impact of illumination as a critical visual characteristic variable on model generalization, we designed our experimental environment as illustrated in Fig. 3(a). Our goal was to create a stable and uniform lighting environment, crucial for ensuring the integrity and reliability of our results. For this purpose, two fill lights were positioned at 45° on either side of the experimental platform. This arrangement created a balanced and even dual light source environment, effectively eliminating potential shadows or irregular illumination during the data-taking process. These fill lights were adjustable, capable of emitting light at three different color temperatures and allowing for controlling the intensity via remote control, thus establishing a uniform dual-source lighting condition.

As an integral part of this setup, we utilize a lux meter (as shown in Fig. 3(b)) to provide precise quantification of the illumination environment. This instrument is crucial for precision measurements of light intensity in various illumination settings, enabling us to describe the attributes of each scene quantitatively. Carefully designed illumination environments and quantitative measurements form the foundation for constructing illumination distributions in our research, ensuring that variations in lighting can be precisely controlled and accurately quantified. All these illumination environment settings built a comprehensive experimental environment with dual controllable light sources, providing a trustworthy platform for our data collection process.

3.3 Data Preparation

Training Set. In order to thoroughly investigate the impact of the visual variable, "illumination", on the generalization capabilities of visual models, we purposefully constructed two training sets. The Full Spectrum Illumination Dataset (FSID) encompasses a range of illumination distributions, incorporating variations in light color and intensity. This representation ensures a uniform distribution of the "illumination" variable in our study. In contrast, the Singular

Illumination Dataset (SID) simulates a narrowed range of illumination by selecting specific median illumination settings from FSID, transitioning from a broad, uniform distribution to a degraded, singular distribution.

Table 1. Illumination settings with different levels of intensity and color temperature, where illumination intensity was measured using the light intensity meter in Fig. 3(b), ensuring an error margin within ±20 lux. And the color temperature was indirectly measured using a gray card under different illumination settings. An average value was calculated from 100 images

Intensity	−2 level	−1 level	0 level	+1 level	+2 level
Warm Light	180 Lux, 3222K	540 Lux, 3812K	900 Lux, 4205K	1260 Lux, 4388K	1620 Lux, 4205K
White Light	200 Lux, 20397K	600 Lux, 15186K	1000 Lux, 12769K	1400 Lux, 12527K	1800 Lux, 11931K
Mixed Light	400 Lux, 8058K	1200 Lux, 7628K	2000 Lux, 7192K	2700 Lux, 6607K	3500 Lux, 6499K

Full Spectrum Illumination Dataset (FSID): For this dataset, we followed a detailed data collection process under various illumination settings. The illumination attributes include light color and illumination intensity. The light color included [Warm, Cool, Mixed] three different attributes, while the illumination intensity was divided into five distinct levels $[-2, -1, 0, +1, 1]$ by measuring the intensity's scope of different light colors. By forming 15 different illumination settings (3 light colors * 5 illumination intensities), we created a series of diverse illumination variations and performed data-taking based on these settings. For each toy, high-resolution images were captured by high resolution camera under these varied illumination settings, ensuring at least 100 clear, unobstructed images from different angles. Eventually, a set of 15,000 images was collected (3 color temperatures * 5 levels of light intensity * 10 categories * 100 images) as our FSID. Figure 4 shows images of Toy 1 under 15 different lighting scenarios as an example.

Singular Illumination Dataset (SID): In this dataset, we focused on constructing image data under a singular illumination setting. We chose the middle illumination setting [Cool light, 0 level] listed in Fig. 4 as the target condition, with at least 1500 images of various poses collected around each recognition object to ensure data volume consistency with FSID. The construction of this dataset served two purposes: on one hand, to explore the impact on visual model generalization when visual representation variables are simplified to a singular distribution; on the other hand, to provide a basis for data augmentation study on enhancing model generalization abilities.

During the data-taking progress of our study, we used the Intel RealSense D435i camera to capture images of 10 toys from a similar height. All RGB images in our two datasets are clear and unobstructed, with a resolution of 640*640. This consistent data-taking method assures that the primary difference between

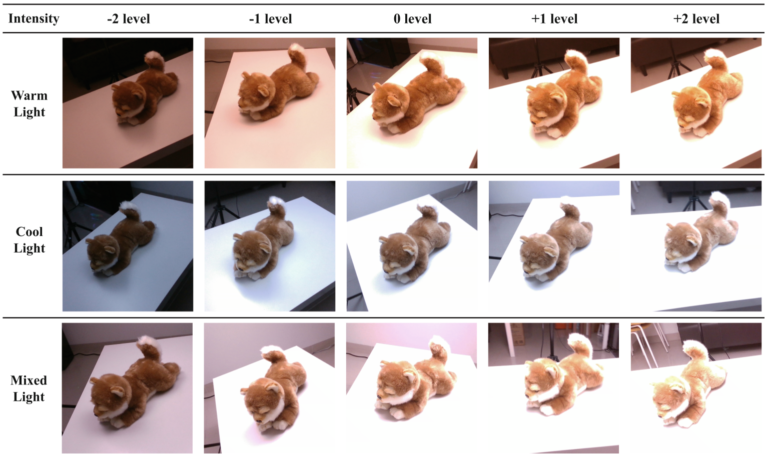


Fig. 4. Toy 1 depicted under 15 illumination settings within the FSID, with light colors [Warm, Cool, Mixed] and intensities $[-2, -1, 0, +1, +2]$. The illumination intensity and color temperature are described in Table 1.

the FSID and SID training sets is attributed to changes in illumination settings, without introducing additional covariates. This provides a solid basis for further analyzing model performance under different illumination settings as a training set, ensuring the reliability of the research findings.

Test Set: Throughout the process of evaluating models' adaptability to changes in illumination, a standardized test set was specifically constructed, characterized by a complexity of lighting variations far exceeding all scenarios within the training sets. The construction of this test set took into account the diverse changes in light color properties and ensured random fluctuations in light intensity across the entire spectrum. Contrasting with the fixed light intensity levels of training set A (five predefined levels for each light color), the test set was designed to cover the complete range of illumination spectrum. Through meticulous adjustments of auxiliary lighting in the testing equipment, a brightness cycle from bright to dark and back to bright was achieved, ensuring a uniform distribution of brightness throughout the range.

To assess the relationship between model generalization capabilities and the illumination distribution in the training set, we constructed a standardized test set with a range of illumination variations that exceed all scenarios in the training set. The construction of this test set considered the diverse changes in light color properties and ensured random fluctuations in light intensity across the entire scope. Contrasting with the fixed light intensity levels of FSID (five predefined levels for each light color), the test set was designed to cover the complete range of illumination intensity scope. By continuously adjusting the light intensity during the data collection process, and cycling from bright to dark and back to bright, we ensured a uniform distribution of light intensity across the entire range in the test set.

3.4 Classification Model

To ensure the experimental findings are widely applicable, we selected a variety of widely-used deep learning models for comparative experiments, including AlexNet [1], VGG [2], ResNet [3], EfficientNet [4], ViT [5], and Swin Transformer [6], including classic CNNs and latest Transformer models. These deep learning models were originally designed to process large, semantically abundant datasets like ImageNet [40], which provides a comprehensive feature collection in visual tasks. However, when applying these deep learning models to the relatively simple training sets in our study, all models find it hard to reach convergence during the training process. This is due to the dominant feature complexity gap between our training set and comprehensive datasets such as ImageNet. Finally, we solved this problem by scaling the models to match the capacity of our training data. For CNN models, including AlexNet, VGG, ResNet, and EfficientNet, we adjusted by reducing the number of channels to one-quarter of their original count. For Transformer architectures, with means ViT and Swin Transformer, we made more meticulous modifications. Specifically, we simplified the number of attention heads per layer and reduced the number of hidden units in each layer, aiming to decrease model complexity and computational burden. All these adjustments simplified the model structures while retaining their powerful image processing capabilities. By optimizing the architectures in this manner, we ensured faster model convergence on smaller-scale datasets, preserving the key architecture of their original designs. After confirming that all deep learning models could be trained on FSID and SID with steady loss reduction and ultimately reached convergence, we completed all preparatory work for the following comparative experiments.

4 Comparative Experiments

4.1 Experiment 1: Uniform Distribution vs. Singular Distribution of Illumination Attributes in Training Set

In our first experiment, we focused on how the distribution of the "illumination" attribute within training datasets affects the generalization performance of deep learning models. Specifically, we utilized the six deep learning models mentioned in Sect. 3.4 and conducted comparative experiments using the Full Spectrum Illumination Dataset (FSID) and the Singular Illumination Dataset (SID) as the training datasets. We aimed to examine the impact of differences in the distribution of the "illumination" variable on model generalization while keeping all other variables unchanged.

To ensure fairness and consistency for all deep learning models, we applied nearly the same training hyperparameters for all experiments. This included setting the data validation split to 0.2, the batch size to 64, choosing Adam as the optimizer, and setting the initial learning rate to 0.001. During the preprocess stage, we follow the procedural of AlexNet [1] by scaling all training images to 224*224 and making color normalization. All models were trained extensively on FSID and SID until their training loss reached convergence. Notably, CNN-based models typically converged within 10 epochs, while Transformer-based models required more training epochs. Specifically, our ViT model needed 20 epochs, and our Swin Transformer model needed 60 epochs to reach the convergence [41]. After completing all training processes, we evaluated the generalization performance of all models trained on FSID and SID using the test set described in Sect. 3.3. The evaluation metrics included model accuracy, precision, and recall on our test set, the experimental results details were shown in Table 2.

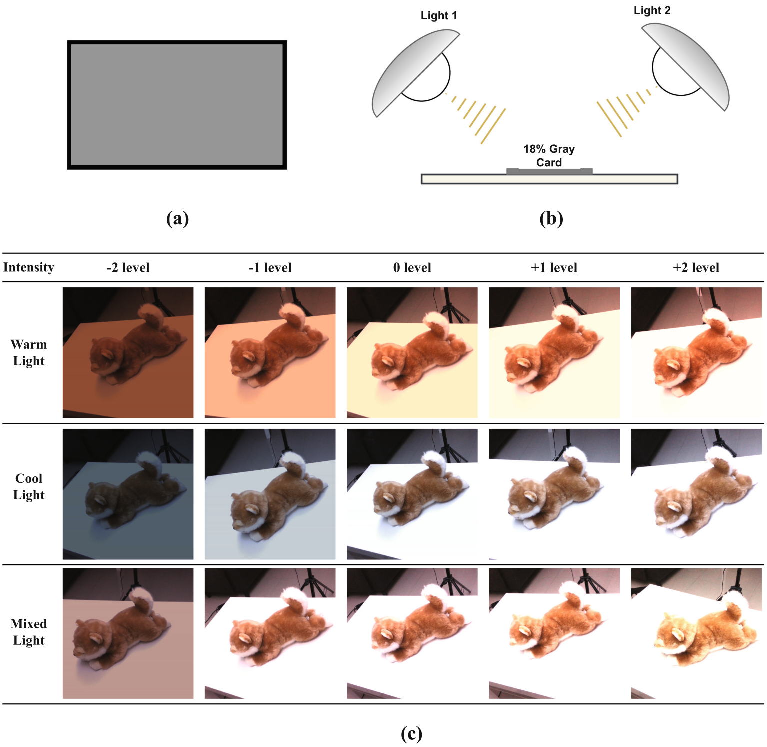
Table 2. Performance comparisons of different models trained on FSID and SID datasets under identical configurations reveal that accuracy declined by 0.67 across the test sets

Metrics	AlexNet			VGG			ResNet			EfficientNet			ViT			Swin_T		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
FSID	0.981	0.982	0.981	0.995	0.995	0.995	0.997	0.997	0.997	0.996	0.996	0.996	0.981	0.981	0.981	0.982	0.982	0.982
SID	0.347	0.580	0.347	0.324	0.570	0.324	0.382	0.313	0.382	0.373	0.678	0.373	0.290	0.604	0.290	0.264	0.502	0.264

By analyzing the results showcased in Table 2, when the illumination distribution degenerates to a singular one, we could clearly recognize the difference of their performance on the test set. Although all models preserved the same training configuration in FSID and SID, the four CNN and two Transformer models exhibited significant performance discrepancies due to their training sets containing different illumination distributions. The SID only contained a singular illumination distribution [Cool light, 0 level] while the FSID provided a full spectrum distribution with 15 illumination settings. Particularly, all models trained on FSID performed great performance on the test set, however, models received about 0.67 drops in accuracy when trained on SID, with similar declines observed in precision and recall. These results compellingly demonstrate that when the visual variable distribution in training sets degenerated (specifically illumination in our study), this led to a catastrophic decrease in their generalization capabilities.

4.2 Experiment 2: Statistical Illumination Vector Mapping Augmentation in the Singular Illumination Dataset

In experiment 1, we confirmed that a singular distribution of illumination significantly reduces the performance of deep learning models. Further observation of the data showed that the color and intensity of lighting have a significant impact on the visual characteristics, especially on the color appearance of training images. Given this observation, we hypothesized that quantifying the pixel-level mapping correlations between different illumination settings could mitigate the decline in model generalization caused by a singular illumination distribution in SID. In experiment 2, we aimed to explore this hypothesis and attempted to address this issue through color channel enhancement methods [42].



(a)

(b)

(c)

Fig. 5. Establishing the illumination settings of FSID to generate extensive illumination vectors for augmenting the SID dataset. (a) 18% gray card, (b) scene assembled for data collection, and (c) images from the SID dataset of Toy 1, enhanced with illumination vectors under diverse illumination settings (detailed in Table 1).

In this experiment, we used an 18% gray card shown in Fig. 5(a) as the subject and replicated the same 15 diverse illumination settings found in the Full Spectrum Illumination Dataset (FSID), as depicted in Fig. 5(b). Under these conditions, we captured multiple photographs. From these, we meticulously selected 100 images with consistent imaging quality for further analysis. For each selected image, we calculated the average values of the R, G, and B color channels under the current lighting conditions, defining it as the environmental illumination vector $V_{\text{ill}}[R, G, B]$, where the light color C includes three types (Warm, Cool, Mixed) and light intensity I is divided into five levels $(-2, -1, 0, +1, +2)$. Then, we calculated the standard illumination environment vector $V_{\text{ill,SID}}[R, G, B]$ in SID for the [Cool, 0 level] scenario and compared it with the vectors $V^{(k)}_{\text{ill,FSID}}[R, G, B]$ under the other 14 different lighting conditions. Based on these ratios, we enhanced 15,000 images in SID, selecting 100 images randomly for each lighting condition for processing. The enhancement process was accomplished by applying the corresponding illumination mapping coefficients to the global pixels of the images, thus creating the Illumination Vector Augmentation Dataset (IVAD).

Figure 5(c) showed the results of Toy1 in IVAD enhanced by illumination mapping enhancement, where the dataset shifted from a singular illumination distribution of [Cool light, 0 level] in SID to 15 types of full-spectrum illumination distributions. This enhancement through color channel-based illumination mapping allowed images in SID, previously constrained by a singular illumination distribution and lacking visual diversity, to present much more complicated visual characteristics. These characteristics were similar to those observed under various illumination settings in FSID. Subsequently, following the setup from Experiment 1, we trained all deep learning models in IVAD. As shown in Table 3, our experimental results indicate that models trained on IVAD achieved an increase of approximately 0.57 in accuracy on the test set compared to those trained on SID. This means that using illumination vector mapping as our data augmentation method could significantly improve models' performance comparing those trained on a singular illumination distribution in SID.

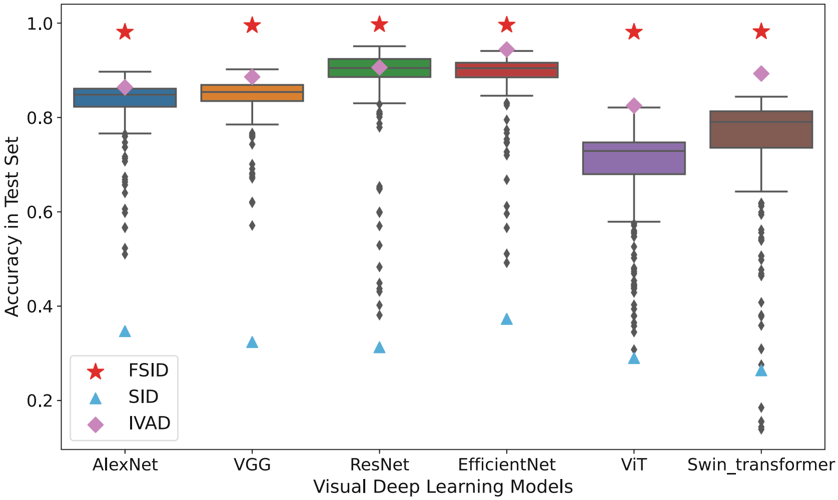


Fig. 6. Each box displayed the optimization of color enhancement parameters achieved through Bayesian optimization using Optuna across six distinct visual models over 200 iterations. It highlighted the progress in model generalization due to color-based data augmentation, showing improvements beyond the IVAD's results from Experiment 2, while still emphasizing the gap in performance compared to the FSID dataset's real-world illumination variations

Table 3. Comparative analysis of model performance on the SID and IVAD, highlighting the effectiveness of illumination vector-based data augmentation method for improving model generalization capabilities

Data	AlexNet			VGG			ResNet			EfficientNet			ViT			Swin_T		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
SID	0.347	0.580	0.347	0.324	0.570	0.324	0.382	0.313	0.382	0.373	0.678	0.373	0.290	0.604	0.290	0.264	0.502	0.264
IVAD	0.864	0.881	0.864	0.886	0.892	0.886	0.906	0.912	0.906	0.944	0.949	0.944	0.825	0.840	0.825	0.893	0.901	0.893

4.3 Experiment 3: Color Augmentation via Bayesian Optimization in the Singular Illumination Dataset

During the training process of deep learning-based visual models, data augmentation techniques play an important role. In Experiment 2 we demonstrated that color channel enhancement could significantly improve model performance. Given that our study mainly focused on the "illumination" attribute, this experiment also focused on color-based data augmentation methods. We utilized Torchvision, a package from the PyTorch [43] project that provides tools and datasets for computer vision. It provided a color augmentation function "torchvision.transforms.ColorJitter" which included 4 variables: brightness, contrast, saturation, and hue. In this section, we aim to maximize the model's performance on the test set by searching for the best parameter configurations on torchvision.transforms.ColorJitter data augmentation function.

To precisely adjust these four color enhancement parameters, we employed the Optuna Bayesian optimization framework [44], with the Tree-structured Parzen Estimator (TPE) as the parameter sampling strategy. During the optimization process, each TPE iteration generated a new set of parameter configurations. These configurations were applied to data augmentation and subsequently used to train models according to the setup established in Experiment 1. After training, we evaluated the effectiveness of each iteration by measuring the model's accuracy on the test set. For six different deep learning models, we conducted 200 optimization iterations for searching the best color augmentation parameter configurations. As shown in Fig. 6, the results demonstrated that after 200 iterations of Bayesian optimization for data augmentation, with the best parameter configurations, the models' performance approached or even surpassed the illumination mapping data augmentation methods used in IVAD. Our experiments showed that color enhancement techniques in Torchvision, fine-tuned through Bayesian optimization, could surpass the illumination vector mapping augmentation method in Experiment 2. However, despite all these improvements, a generalization gap still existed when compared to the FSID dataset, which included actual illumination changes. This emphasized the irreplaceable of visual feature complexity in real datasets for constructing a robust model generalization.

Table 4. Performance analysis of models across different distributions and augmentation methods within FSID, SID, IVAD, and Bayesian Optimization Data Augmentation (BO-DA). It demonstrates that while data augmentation methods (IVAD and BO-DA) significantly improved models' generalization, a notable gap persists when compared to FSID which took data from a real-world illumination distribution. The gap highlights that artificial illumination variations introduced through data augmentation inherently involve certain generalization limitations

Metrics	AlexNet			VGG			ResNet			EfficientNet			ViT			Swin_T		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
FSID	**0.981**	**0.982**	**0.981**	**0.995**	**0.995**	**0.995**	**0.997**	**0.997**	**0.997**	**0.996**	**0.996**	**0.996**	**0.981**	**0.982**	**0.981**	**0.982**	**0.983**	**0.982**
SID	0.347	0.580	0.347	0.324	0.570	0.324	0.382	0.313	0.382	0.373	0.678	0.373	0.290	0.604	0.290	0.264	0.502	0.264
IVAD	0.864	0.881	0.864	0.886	0.892	0.886	0.906	0.912	0.906	0.944	0.949	0.944	0.825	0.840	0.825	0.893	0.901	0.893
BO-DA	0.897	0.900	0.897	0.902	0.908	0.902	0.951	0.952	0.951	0.941	0.943	0.941	0.821	0.822	0.821	0.844	0.855	0.844

5 Conclusion

To conclude our study, we summarized all results in Table 4. In our research, we focused on the visual representation variable "illumination" by forming the Full Spectrum Illumination Dataset (FSID) with uniform distribution and the Singular Illumination Dataset (SID) with singular distribution on datasets of a classification task. In Experiment 1, we proved that when a visual representation variable degenerated to a singular distribution, it will occur a catastrophic decline in deep learning-based visual models. We performed an illumination vector mapping data augmentation method in Experiment 2. Models' generalization abilities trained by the Illumination Vector Augmentation Dataset (IVAD) had significant improvements. Experiment 3 further conducted a Bayesian Optimization Data Augmentation (BO-DA) method, which slightly outperformed the models' performance trained with IVAD.

Nevertheless, whether employing intuitive color mapping techniques or implementing color-based data augmentation through Bayesian optimization, these strategies still exhibit a generalization gap when compared to models trained on a complex, real-world illumination dataset. This outcome highlights that while data augmentation can enhance model generalization, it has inherent limitations. It is essential to ensure that different visual representation variables are sufficiently complex and diverse. Most importantly, any visual representation variable should not degrade into a singular distribution. Proper dataset design is critical for achieving robust model generalization, emphasizing the importance of incorporating diverse visual features.

6 Data Limitation

One limitation of our study is the image format. Currently, images were saved in PNG format using an Intel RealSense D435i camera, which does not preserve physical illumination information as effectively as RAW format. RAW captures linear data directly from the sensor, crucial for accurate color correction and augmentation. We acknowledge that using PNG may affect our results and plan to use RAW format in future work to enhance data accuracy and robustness.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
3. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
4. Tan, M., Le, Q.: Efficientnet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR (2019)
5. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
6. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
7. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
8. Ren, S., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
9. Redmon, J., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
12. Xu, S., et al.: Computer vision techniques in construction: a critical review. Arch. Comput. Methods Eng. **28**, 3383–3397 (2021)
13. Esteva, A., et al.: Deep learning-enabled medical computer vision. NPJ Digit. Med. **4**(1), 5 (2021)
14. Chang, M.-C., et al.: AI city challenge 2020-computer vision for smart transportation applications. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2020)
15. Kuznetsova, A., et al.: The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale. Int. J. Comput. Vis. **128**(7), 1956–1981 (2020)
16. Bian, Y., Chen, H.: When does diversity help generalization in classification ensembles? IEEE Trans. Cybern. **52**(9), 9059–9075 (2021)
17. Ekstrom, M.P.: Digital Image Processing Techniques, vol. 2. Academic Press (2012)
18. Zhang, H., et al.: Mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)

19. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
20. Li, C., et al.: YOLOv6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
21. 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 (2023)
22. Sun, P., et al.: Scalability in perception for autonomous driving: waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
23. Yoneda, K., et al.: Automated driving recognition technologies for adverse weather conditions. IATSS Res. **43**(4), 253–262 (2019)
24. Tremblay, M., et al.: Rain rendering for evaluating and improving robustness to bad weather. Int. J. Comput. Vis. **129**, 341–360 (2021)
25. DeVries, T., Taylor, G.W.: Dataset augmentation in feature space. arXiv preprint arXiv:1702.05538 (2017)
26. Wang, Y., et al.: Regularizing deep networks with semantic data augmentation. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3733–3748 (2021)
27. Suh, S., Lukowicz, P., Lee, Y.O.: Discriminative feature generation for classification of imbalanced data. Pattern Recogn. **122**, 108302 (2022)
28. Hnewa, M., Radha, H.: Object detection under rainy conditions for autonomous vehicles: a review of state-of-the-art and emerging techniques. IEEE Signal Process. Mag. **38**(1), 53–67 (2020)
29. Wen, Y., et al.: Combining ensembles and data augmentation can harm your calibration. In: International Conference on Learning Representations (2021)
30. Li, H., et al.: Domain generalization for medical imaging classification with linear-dependency regularization. In: Advances in Neural Information Processing Systems, vol. 33, pp. 3118–3129 (2020)
31. Zhao, Y., et al.: Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
32. Zhang, L., et al.: Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. IEEE Trans. Med. Imaging **39**(7), 2531–2540 (2020)
33. Mancini, M., Akata, Z., Ricci, E., Caputo, B.: Towards recognizing unseen categories in unseen domains. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 466–483. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_28
34. Mancini, M., et al.: Robust place categorization with deep domain generalization. IEEE Robot. Autom. Lett. **3**(3), 2093–2100 (2018)
35. Taylor, L., Nitschke, G.: Improving deep learning with generic data augmentation. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE (2018)
36. Wu, R., et al.: Deep image: scaling up image recognition. arXiv preprint arXiv:1501.02876, **7**(8), 4 (2015)
37. Takahashi, R., Matsubara, T., Uehara, K.: Data augmentation using random image cropping and patching for deep CNNs. IEEE Trans. Circuits Syst. Video Technol. **30**(9), 2917–2931 (2019)
38. Varior, R.R., et al.: Learning invariant color features for person reidentification. IEEE Trans. Image Process. **25**(7), 3395–3410 (2016)

39. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
40. Deng, J., et al.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2009)
41. Xu, P., et al.: Optimizing deeper transformers on small datasets. In: Proceedings of the ACL-IJCNLP 2021 (2021)
42. Ancuti, C.O., et al.: Color channel compensation (3C): a fundamental preprocessing step for image enhancement. IEEE Trans. Image Process. **29**, 2653–2665 (2019)
43. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
44. Akiba, T., et al.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)

Harmonizing Regression-Classification Inconsistency for Task-Specific Decoupling in Underwater Object Detection

Minrui Xiang, Tianyang Xu, and Xiaojun Wu(✉)

School of Artificial Intelligence and Computer Science,
Jiangnan University, Wuxi, China
6213160023@stu.jiangnan.edu.cn,
{tianyang.xu,wu_xiaojun}@jiangnan.edu.cn

Abstract. In the realm of underwater object detection, challenging conditions, characterized by blurriness and complexity, suppress the representative power of general deep backbone features. In this work, we propose an innovative decoupled head structure building upon the YOLOv7 framework. This structure segregates classification and regression branches to better capture the semantic features required for each subtask. In order to make effective use of the contradiction between classification and regression, we introduce the adjacent feature layer as a complementary operator to harmonizing subtasks. To address the issue of image blurriness in the underwater environment, a probabilistic modeling approach was adopted for regression box handling. The ultimate detection outcome is determined jointly by the classification and regression branches, enhancing the overall consistency of the category and bounding box results. An additional branch is introduced within the classification branch and seamlessly integrated to further augment the coherence of the detection outcomes. This comprehensive approach effectively addresses the challenges posed by the underwater environment, significantly improving the accuracy and robustness of underwater object detection.

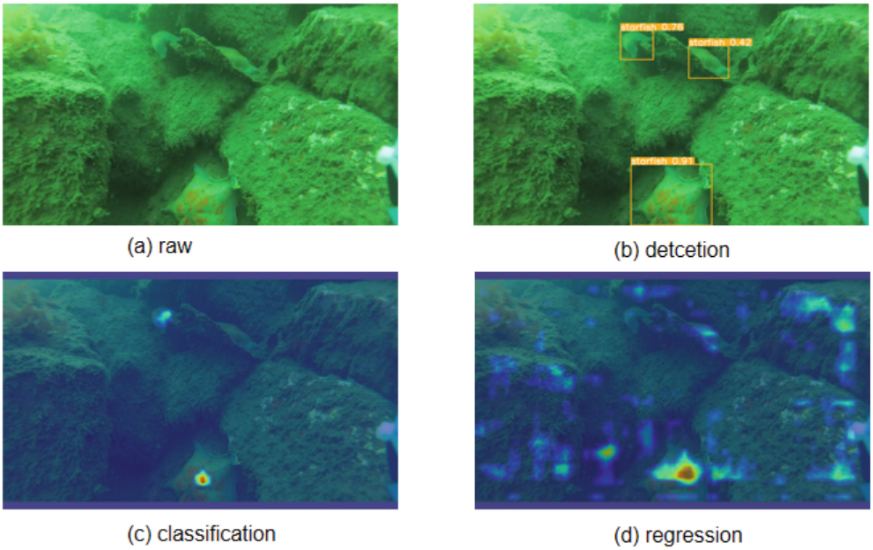
Keywords: Underwater Object Detection · YOLOv7 · Decoupled Head Structure · Task-Specific

1 Introduction

In recent years, the rapid development of the field of computer vision has brought great potential for various applications, especially in object detection tasks [8,11]. As an important branch in the field of computer vision, underwater object detection (UOD) has a wide range of applications, including underwater resource exploration, environmental monitoring, underwater robot operation and so on. Despite notable advancements in the broader field of object detection, the

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 206–218, 2025.
https://doi.org/10.1007/978-3-031-78444-6_14

intricate and unique nature of the underwater environment presents persistent challenges for underwater object detection in real marine conditions [1,5,19].



(a) raw

(b) detcetion

(c) classification

(d) regression

Fig. 1. Utilizing Grad-CAM visualization for the classification and regression output.

The challenges in underwater object detection stem from the unique characteristics of the underwater environment and the complexities involved in achieving effective detection within such conditions. These challenges are influenced by a combination of optical properties, image degradation, fluctuations in illumination, and underwater noise. The optical properties of water lead to color distortion and uneven brightness in underwater images, resulting in targets with an unstable appearance. Image degradation, including blurring, scattering, and interference from particles, further complicates the identification of target edges and details. Changes in illumination introduce variations in brightness and contrast, which pose additional challenges in distinguishing visual features of targets. Moreover, underwater noise, comprising suspended particles and wave-induced disturbances, combines with targets, reducing their discriminative characteristics. These factors collectively contribute to a deterioration in the quality of underwater images, exacerbating the inherent challenges associated with underwater object detection.

In the previous research [10,12,20,21] on underwater target detection, underwater image enhancement is widely used, relevant studies [5,28] have identified that underwater image enhancement may potentially inhibit the performance of object detection. Particularly, in complex scenarios, image enhancement might exacerbate the suppression of detector performance, potentially increasing interference caused by the background. Despite the capability of underwater image enhancement to mitigate degradation issues in underwater images and obtain visually improved images, it introduces additional quality degradation problems, adversely affecting underwater target detection. Although underwater color cast is not a primary interference factor, its introduction of diverse colors through enhancement may impact the performance of the detector.

Unlike prior methods, we harness the robust feature extraction capabilities of YOLOv7 [26] and refine its head section. Various studies [24,29,30,34] indicate fluctuating demands for semantic features across detection subtasks, prompting the adoption of a decoupled head structure by many detectors [9,13,16,31]. Visualizing the output of this decoupled head (as illustrated in Fig. 1) offers insight into the focal points of the classification and regression branches in underwater object detection tasks. Specifically, the classification branch predominantly targets the central aspect of the object, whereas the regression branch emphasizes the object's surroundings. Decoupling the head facilitates precise adjustments, aligning both branches with the unique characteristics of underwater scenes. While the classification branch concentrates on the object's internal structure and shape, the regression branch attends to its positional information and external environment. This independent handling of focus areas optimizes the performance of both classification and regression tasks. We implement head decoupling within YOLOv7 [26], enabling detection subtasks to focus on their requisite semantic content. Additionally, we introduce a Task-Specific Enhancement Module, integrating information from nearby feature layers to bolster subtasks. To address object blurring in underwater scenarios, we employ a probabilistic modeling approach where the regression branch outputs probabilities for each interval, culminating in the expected value of the interval for final localization. Finally, our Regression Guidance Module enhances consistency among subtasks, thereby refining detector performance in complex scenarios.

The main contributions can be summarized as follows:

- We proposed a decoupled head structure built upon YOLOv7, where the classification and regression branches no longer share the same feature input. This design allows each subtask to concentrate on its specific semantic information, enhancing the model's ability to address underwater object detection challenges.
- We enhanced the input to the head by introducing neighboring feature layer information tailored to the semantic requirements of each subtask.
- We adopted a probabilistic modeling approach for regression box probability, where the model's regression output represents the likelihood at various positions. This effectively addresses the issue of underwater object blurriness.
- We introduced supervision from the regression branch into the classification branch, enhancing the consistency between the detector's regression and classification outputs.

2 Related Work

We will review the pertinent technique related to our work briefly in this section. The YOLOv7 model, developed by Chien-Yao Wang and Alexey Bochkovskiy et al. in 2022, introduces enhancements to achieve an effective balance between detection efficiency and precision. Comprising four modules, namely the Input module, Backbone network, Head network, and Prediction network, YOLOv7 incorporates strategies such as E-ELAN and model scaling.

Input module: Utilizing mosaic and hybrid data enhancement techniques, the Input module ensures uniform scaling of color images to a 640 × 640 size, meeting the backbone network's input size requirements.

Backbone network: Consisting of CBS, E-ELAN, and MP1, the Backbone network employs convolution, batch normalization, and SiLU activation in the CBS module. E-ELAN enhances learning ability, and MP1 facilitates channel and size reduction, enhancing feature extraction.

Head network: Adopting the Feature Pyramid Network architecture with PANet design, the Head network integrates CBS blocks, Sppcspc structure, E-ELAN, and MP2. These elements improve perceptual field and feature extraction ability.

Prediction network: Employing a Rep structure, the Prediction network adjusts image channels and utilizes 1×1 convolution for confidence, category, and anchor frame predictions. Inspired by RepVGG, the Rep structure simplifies to a basic convolution for practical predictions, reducing network complexity without compromising performance. Prediction network: Employing a Rep structure, the Prediction network adjusts image channels and utilizes 1×1 convolution for confidence, category, and anchor frame predictions. Inspired by RepVGG, the Rep structure simplifies to a basic convolution for practical predictions, reducing network complexity without compromising performance. Prediction network: Employing a Rep structure, the Prediction network adjusts image channels and utilizes 1×1 convolution for confidence, category, and anchor frame predictions. Inspired by RepVGG, the Rep structure simplifies to a basic convolution for practical predictions, reducing network complexity without compromising performance (Fig. 2).

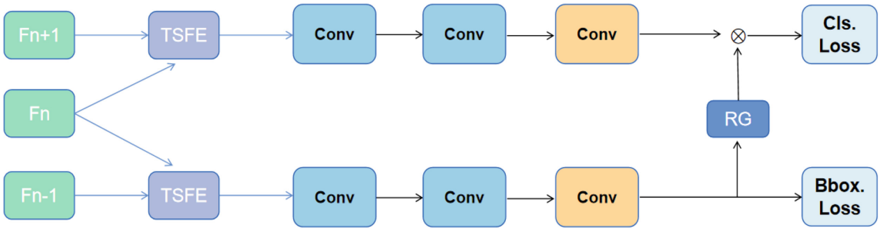


Fig. 2. This is the pipeline: we introduce neighboring layer feature information F_{n+1} and F_{n-1}, incorporating Task-Specific Feature Enhancement modules into both the classification and regression branches. The regression branch adopts a probabilistic modeling approach, outputting interval probabilities. Finally, the regression-guided module learns to combine this distribution with the results from the classification branch.

3 Methodology

Classification and regression are two highly correlated yet contradictory tasks in the object detection. In the underwater environment, the extracted features are abundant but lack richness in information. Leveraging this characteristic, in this paper, we use YOLOv7 as the baseline and propose a novel decoupled head structures called TSD-YOLO shown in Fig. 1. By taking advantage of the relationship between subtasks, we aim to better handle feature information, addressing the challenges posed by the complex underwater environment.

3.1 Decouple Head

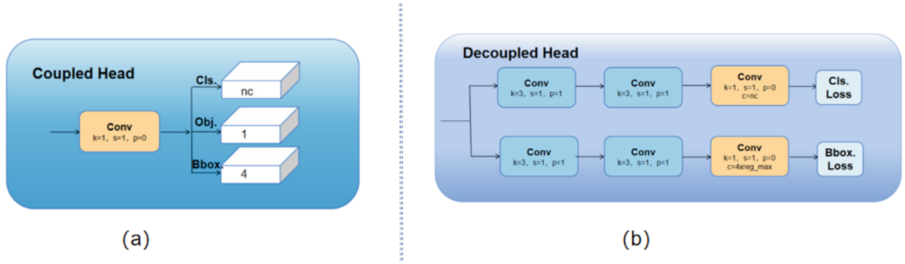


Fig. 3. Frameworks of Couple Head and Decouple Head.

YOLOv7, positioned as a cutting-edge algorithm, excels in object detection tasks due to its refined network architecture and advanced training strategies, showcased in Fig. 3(a). Despite the traditionally shared feature maps for classification and regression subtasks, recent studies reveal distinctive preferences between these tasks. In challenging underwater environments, the conventional coupled head structure may not effectively handle extracted features.

To address this, we propose a novel decoupled head structure (Fig. 3(b)). The two layers of blue modules employ 3 × 3 convolution operations, capturing diverse information within the feature map. The yellow module, a 1 × 1 convolution operation, synthesizes the final result based on learned information. This innovative structure tailors to the nuances of underwater object detection, segregating and processing information effectively. Our approach aims to enhance adaptability to underwater features, improving overall detection accuracy and robustness.

3.2 Task-Specific Feature Enhancement

In the feature pyramid structure, lower-level feature maps with higher resolution excel in capturing image details for precise target localization, while higher-level

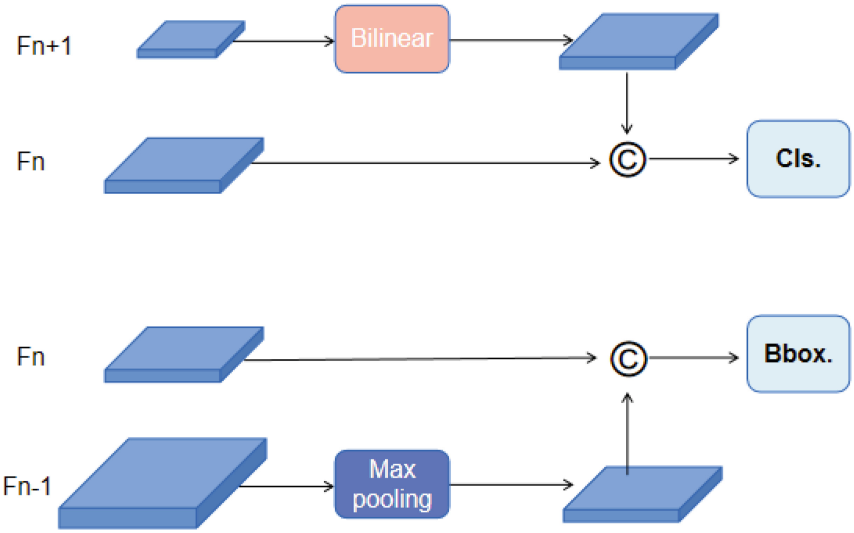


Fig. 4. Task-Specific Feature Enhancement

feature maps capture abstract semantic information crucial for target classification. To enhance effective feature utilization (as shown in Fig. 4), we propose a Task-Specific Feature Enhancement Module.

For the classification branch, we introduce an additional high-level feature map, enlarged through bilinear interpolation, and concatenate it with the original feature layer. In the regression branch, a lower-level feature map is incorporated, reducing its size through max pooling, preserving local structures and textures better than average pooling.

Dedicated information introduction for each subtask reinforces specific feature information required, optimizing the fusion of detailed local information and abstract semantic context. This modular approach contributes to superior performance in both target localization and classification.

3.3 Probabilistic Modeling

The term "probability modeling" refers to the introduction of a more versatile probability distribution for representing bounding boxes. Unlike traditional box regression methods that often rely on a deterministic Dirac distribution, which assumes a highly certain target position, this approach embraces a more general probability distribution. This distribution accommodates the uncertainty associated with bounding box representations in complex scenes, where targets may have multiple possible positions or exhibit boundary fuzziness. The adoption of a more general probability distribution enhances the model's flexibility, enabling it to better adapt to uncertainties in target positions across diverse scenarios. By incorporating this versatile probability modeling, the model becomes more adept at handling uncertainties, thereby improving the robustness of box regression in complex and challenging environments.

3.4 Regression Guidance

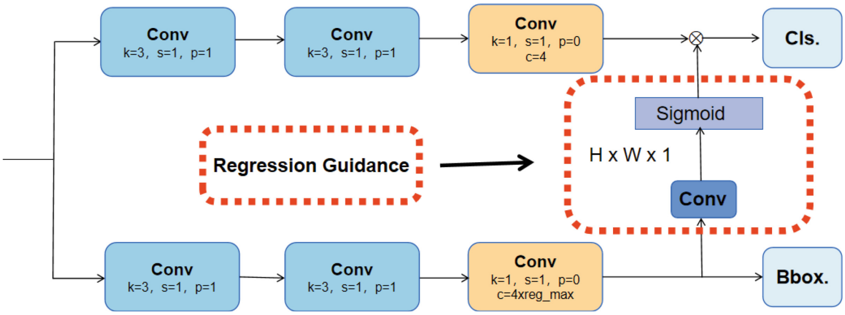


Fig. 5. Regression Guidance

Classification and regression tasks often share the same feature representation. In deep learning models for object detection, it is common to use shared Convolutional Neural Network (CNN) layers to extract features from images, which are then utilized for both classification and regression tasks. Consistency between classification and regression tasks can be achieved by introducing a branch that connects to the regression branch. This branch serves as an auxiliary task, guiding the classification task indirectly by supervising the learning of the regression branch. This design of consistency ensures that the model learns a coherent feature representation for both tasks, enhancing the overall performance of the model. Therefore, we have devised a Regression Guidance Module, which improves the consistency of sub-tasks by learning the preceding probability distribution (Fig. 5).

4 Experiments

We objectively assess TSD-YOLO through qualitative experiments, gauging its performance. A series of ablation experiments is systematically conducted to evaluate the influence of different network structures and modules in our investigation.

4.1 Experimental Scheme

Our model underwent an assessment using the DUO [18] dataset. This recently released underwater dataset presents a diverse array of underwater scenes along with more sensible annotations. A refined iteration of the UTDAC2020 dataset

[3], the DUO dataset is a product of enhancements originating from the 2020 Underwater Target Detection Algorithm Competition. Comprising a total of 7,782 images (6,671 for training and 1,111 for testing), the dataset includes 74,515 instances across four primary categories: echinus, holothurian, starfish, and scallop. The images are provided in four distinct resolutions: 3840 × 2160, 1920 × 1080, 720 × 405, and 586 × 480. Within this framework, we conducted a thorough evaluation of our model's performance specifically on the DUO dataset.

We employ multi-scale training, configuring the long edge to 640 and the short edge to 640 to avoid repetition. The training regimen spans 300 epochs with an initial learning rate of 0.0025, subject to a 0.1 decay rate. Our method is trained on a single GeForce RTX 2080Ti GPU, utilizing a total batch size of 8 during training. We opt for SGD as the training optimization algorithm, setting the weight decay to 0.0001 and the momentum to 0.9. Throughout the experiments, traditional horizontal flipping is the only data augmentation applied.

4.2 Qualitative Results

We conducted comparisons with other state-of-the-art methods and applied our approach to various sizes of both YOLOv5 and YOLOv7. The results are presented in Table 1 and Table 2.

Table 1. Comparison with the state-of-the-art methods of generic detectors and underwater detectors on the DUO dataset

Methods	AP	AP50	AP75	echinus	starfish	holothurian	scallop
Generic Object Detector:							
Faster R-CNN [23]	61.3	81.9	69.5	70.4	71.4	61.4	41.9
Cascade R-CNN [2]	61.2	82.1	69.2	69.0	72.0	61.9	41.9
AutoAssign [32]	66.1	85.7	72.6	74.1	75.5	65.8	48.9
SABL w/ Cascade R-CNN [27]	63.4	81.2	70.5	72.0	74.0	64.7	42.8
DetectoRS [22]	64.8	83.5	72.4	73.5	74.3	65.8	45.7
Deformable DETR [33]	63.7	84.4	71.9	71.6	73.9	63.0	46.3
GFL [14]	65.5	83.7	71.9	74.2	75.9	64.3	47.5
YOLOv7 [26]	68.0	88.0	75.9	75.6	76.3	68.0	52.1
Underwater Object Detector:							
ROIMIX [17]	61.9	81.3	69.9	70.7	72.4	63.0	41.7
ERL-Net [7]	64.9	82.4	73.2	71.0	74.8	67.2	46.5
Boosting R-CNN [25]	63.5	78.5	71.1	69.0	74.5	63.8	46.8
SWIPENet [4]	63.0	79.7	72.5	68.5	73.6	64.0	45.9
RoIAttn [15]	62.3	82.8	71.4	70.6	72.6	63.4	42.5
GCC-Net [6]	69.1	87.8	76.3	75.2	76.7	68.2	56.3
TSD-YOLO	**70.6**	89.3	77.4	77.1	77.8	69.1	53.4

We evaluate the performance of TSD-YOLO through comparisons with the baseline method YOLOv7 [26] and state-of-the-art methods, categorizing them into generic object detectors (GOD) and underwater object detectors (UOD). Table 1 presents a summary of the results on the DUO dataset, revealing two key observations. Firstly, in terms of AP accuracy, TSD-YOLO achieves a notable 70.6 AP. In comparison with the baseline method YOLOv7, our proposed method outperforms YOLOv7 by 2.6% (68.0% vs 70.6%). To provide further context, TSD-YOLO exhibits significant performance advantages over DetectoRS by 5.8% (64.8% vs 70.6%), Deformable DETR by 56.9% (63.7% vs 70.6%), GFL by 5.1% (65.5% vs 70.6%), and AutoAssign by 4.5% (66.1% vs 70.6%). These results underscore a substantial margin of improvement achieved by TSD-YOLO across various methodologies, highlighting its effectiveness in both generic and underwater object detection scenarios. Regarding the comparison with UOD methods, we have selected recent open-source approaches to assess the performance of TSD-YOLO. Our proposed method stands out as the top-performing model among these methods. Specifically, TSD-YOLO surpasses SWIPENET by 6.1% (63.0% vs 69.1%), Boosting R-CNN by 5.6% (63.5% vs 69.1%), RoIAttn by 6.8% (62.3% vs 69.1%), and ERL-Net by 4.2% (64.9% vs 69.1%).

TSD-YOLO effectively addresses these challenges, enhancing the visibility of objects in low-contrast regions and thereby significantly improving the per-

Table 2. Comparison of detection performance for different YOLO versions and their depth-wise variants.

Method	Precision	Recall	mAP_0.5	mAP_0.5:0.95
YOLOv5n	84.0	72.1	80.2	55.4
YOLOv5s	88.1	75.5	83.8	61.8
YOLOv5m	88.7	76.6	84.6	64.6
YOLOv5l	87.8	78.2	85.0	66.4
YOLOv7n	84.3	73.1	81.9	56.8
YOLOv7s	88.6	75.6	84.6	62.5
YOLOv7m	89.0	77.3	85.2	66.2
YOLOv7l	88.1	78.7	86.9	68.0
TSD-YOLOv5n	82.3	69.1	77.8	57.1
TSD-YOLOv5s	87.2	74.4	84.4	63.5
TSD-YOLOv5m	84.2	78.9	86.1	66.1
TSD-YOLOv5l	84.7	80.1	87.1	68.5
TSD-YOLOv7n	82.6	69.7	82.9	58.4
TSD-YOLOv7s	88.7	75.0	85.2	64.5
TSD-YOLOv7m	85.3	79.1	87.9	68.9
TSD-YOLOv7l	85.6	81.2	89.3	70.6

formance of UOD tasks by mitigating the difficulties in feature extraction and enriching the information contained in the extracted features. Moreover, it's essential to note that both YOLOv5 and YOLOv7 feature coupled head structures. To showcase the effectiveness of our proposed method, we implemented it on both these models, each configured with different sizes. The results, presented in Table 2, unequivocally highlight the impact of our method.

The data clearly indicates that our approach yields an average improvement of two points across both YOLOv5 and YOLOv7. Particularly noteworthy is the discernible impact on YOLOv7, known for its more efficient architecture compared to YOLOv5. This efficiency translates into a higher enhancement effect, underscoring the adaptability and potency of our method in optimizing different model architectures.

These findings contribute to a robust validation of our method's versatility, demonstrating its capability to enhance object detection across varying model configurations and sizes, ultimately affirming its utility in diverse applications.

4.3 Ablation Study

Table 3. Differential ablation experiments of each module.

Baseline	DH	TSFE	PM	RG	mAP
✓					68%
✓	✓				69.2%
✓	✓	✓			70.2%
✓	✓	✓	✓		70.4%
✓	✓	✓	✓	✓	70.6%
✓	✓	✓			70.2%
✓	✓		✓		69.4%
✓	✓			✓	69.3%

In this section, we conducted a series of ablation experiments on the DUO dataset to assess the efficacy of each module pertaining to the four innovation points. The results, as depicted in the table, reveal that DH (Decouple Head) exhibits the most substantial enhancement for the model, achieving an improvement of approximately 1.2%. Following closely, TSFE (Task-Specific Feature Enhancement) contributes around 1.0%, while the improvements associated with the PM (Probabilistic Modeling) and RG (Regression Guidance) modules are approximately 0.2% each. Furthermore, under the decoupling condition, the enhancements for the three modules are 1%, 0.2%, and 0.1%, respectively. Notably, the incorporation of additional neighboring layer information significantly contributes to the overall improvement of the model (Table 3).

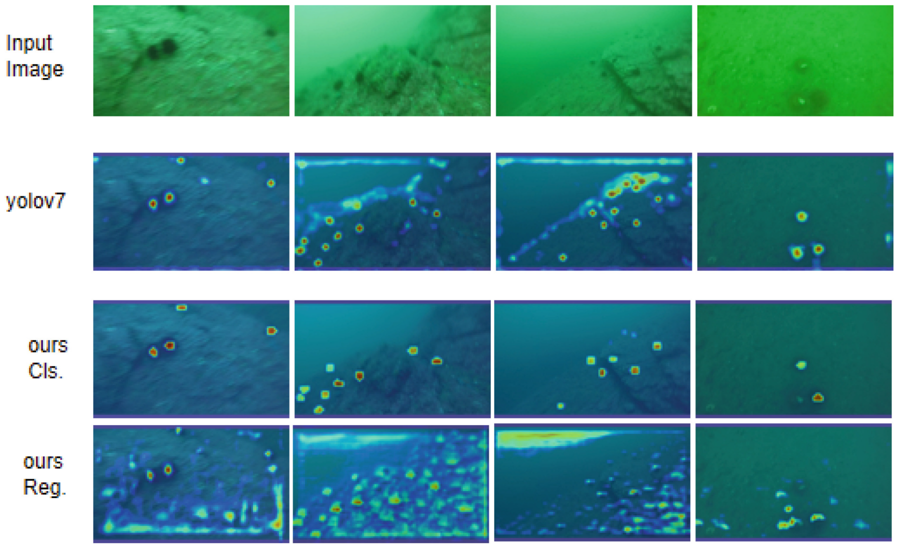


Fig. 6. Comparing visualization results with the baseline using Grad-CAM

4.4 Visualization

We selected low-quality images from the DUO dataset for visualization analysis. It is evident that the feature visualization images generated by YOLOv7 encapsulate both classification and regression information, leading to potential interference with the detection results. In contrast, our approach produces feature visualization images that distinctly focus on regression and classification information. This refinement allows us to obtain more precise feature information. The synergistic integration of these two aspects significantly enhances the overall accuracy of the detection results TSD-YOLO demonstrates superior handling of features (Fig. 6).

5 Conclusion

In this paper, we propose a Task-Specific Decoupling YOLO (TSD-YOLO) to address the challenge of extracting features from low-quality images in underwater environments. Our approach originates from the relationship between the two detection subtasks, classification and regression. We employ a decoupling structure based on contradictions and further introduce additional information to enhance each subtask. Additionally, we propose regression-guided classification from a consistency perspective. Finally, we adopt a probabilistic modeling approach to address challenges posed by the underwater environment.

Acknowledgments. This work was supported in part by the National Key R&D Program of China (2023YFE0116300), and in part by the National Natural Science Foundation of China (62020106012, 62106089, 62336004, 62332008) .

References

1. Anwar, S., Li, C.: Diving deeper into underwater image enhancement: a survey. Signal Process. Image Commun. 115978 (2020). https://doi.org/10.1016/j.image.2020.115978
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154–6162 (2018)
3. Chen, L., et al.: Swipenet: object detection in noisy underwater images. Cornell University - arXiv (2020)
4. Chen, L., et al.: Swipenet: object detection in noisy underwater scenes. Pattern Recogn. **132**, 108926 (2022)
5. Chen, X., Lu, Y., Wu, Z., Yu, J., Wen, L.: Reveal of domain effect: how visual restoration contributes to object detection in aquatic scenes. arXiv preprint arXiv:2003.01913 (2020)
6. Dai, L., Liu, H., Song, P., Liu, M.: A gated cross-domain collaborative network for underwater object detection. arXiv preprint arXiv:2306.14141 (2023)
7. Dai, L., Liu, H., Song, P., Tang, H., Ding, R., Li, S.: Edge-guided representation learning for underwater object detection. arXiv preprint arXiv:2306.00440 (2023)
8. Fan, D.P., Ji, G.P., Xu, P., Cheng, M.M., Sakaridis, C., Van Gool, L.: Advances in deep concealed scene understanding. Vis. Intell. **1**(1), 16 (2023)
9. Ge, Z., Liu, S., Li, Z., Yoshie, O., Sun, J.: OTA: optimal transport assignment for object detection. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021). https://doi.org/10.1109/cvpr46437.2021.00037
10. Islam, M.J., Xia, Y., Sattar, J.: Fast underwater image enhancement for improved visual perception. IEEE Robot. Autom. Lett. **5**(2), 3227–3234 (2020)
11. Jia, Z., Sun, S., Liu, G., Liu, B.: MSSD: multi-scale self-distillation for object detection. Vis. Intell. **2**(1), 8 (2024)
12. Jiang, Z., Li, Z., Yang, S., Fan, X., Liu, R.: Target oriented perceptual adversarial fusion network for underwater image enhancement. IEEE Trans. Circuits Syst. Video Technol. **32**(10), 6584–6598 (2022)
13. Li, X., et al.: Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. Cornell University - arXiv (2020)
14. Li, X., et al.: Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. In: Advances in Neural Information Processing Systems, vol. 33, pp. 21002–21012 (2020)
15. Liang, X., Song, P.: Excavating ROI attention for underwater object detection. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 2651–2655. IEEE (2022)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017). https://doi.org/10.1109/iccv.2017.324
17. Lin, W.H., Zhong, J.X., Liu, S., Li, T., Li, G.: Roimix: proposal-fusion among multiple images for underwater object detection. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ICASSP 2020, pp. 2588–2592. IEEE (2020)
18. Liu, C., et al.: A dataset and benchmark of underwater object detection for robot picking. Cornell University - arXiv (2021)
19. Liu, R., Fan, X., Zhu, M., Hou, M., Luo, Z.: Real-world underwater enhancement: challenges, benchmarks, and solutions under natural light. IEEE Trans. Circuits Syst. Video Technol. **30**(12), 4861–4875 (2020)

20. Liu, R., Jiang, Z., Yang, S., Fan, X.: Twin adversarial contrastive learning for underwater image enhancement and beyond. IEEE Trans. Image Process. **31**, 4922–4936 (2022)
21. Mu, P., Qian, H., Bai, C.: Structure-inferred bi-level model for underwater image enhancement. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 2286–2295 (2022)
22. Qiao, S., Chen, L.C., Yuille, A.: Detectors: detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10213–10224 (2021)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
24. Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020). https://doi.org/10.1109/cvpr42600.2020.01158
25. Song, P., Li, P., Dai, L., Wang, T., Chen, Z.: Boosting R-CNN: reweighting R-CNN samples by RPN's error for underwater object detection. Neurocomputing **530**, 150–164 (2023)
26. 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)
27. Wang, J., et al.: Side-aware boundary localization for more precise object detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 403–419. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_24
28. Wang, Y., et al.: Is underwater image enhancement all object detectors need? IEEE J. Oceanic Eng. (2023)
29. Wu, Y., et al.: Rethinking classification and localization for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10186–10195 (2020)
30. You, J.: Deep neural networks for object detection. Highlights in Science, Engineering and Technology, pp. 159–165 (2022). https://doi.org/10.54097/hset.v17i.2576
31. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020). https://doi.org/10.1109/cvpr42600.2020.00978
32. Zhu, B., et al.: Autoassign: differentiable label assignment for dense object detection. arXiv preprint arXiv:2007.03496 (2020)
33. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
34. Zhuang, J., Qin, Z., Yu, H., Chen, X.: Task-specific context decoupling for object detection (2023)

S3Simulator: A Benchmarking Side Scan Sonar Simulator Dataset for Underwater Image Analysis

S. Kamal Basha and Athira Nambiar(✉)

Department of Computational Intelligence, Faculty of Engineering and Technology,
SRM Institute of Science and Technology, Kattankulathur 603203, Tamil Nadu, India
{c58527,athiram}@srmist.edu.in

Abstract. Acoustic sonar imaging systems are widely used for underwater surveillance in both civilian and military sectors. However, acquiring high-quality sonar datasets for training Artificial Intelligence (AI) models confronts challenges such as limited data availability, financial constraints, and data confidentiality. To overcome these challenges, we propose a novel benchmark dataset of Simulated Side-Scan Sonar images, which we term as **'S3Simulator dataset'**. Our dataset creation utilizes advanced simulation techniques to accurately replicate underwater conditions and produce diverse synthetic sonar imaging. In particular, the cutting-edge AI segmentation tool i.e. Segment Anything Model (SAM) is leveraged for optimally isolating and segmenting the object images, such as ships and planes, from real scenes. Further, advanced Computer-Aided Design tools i.e. SelfCAD and simulation software such as Gazebo are employed to create the 3D model and to optimally visualize within realistic environments, respectively. Further, a range of computational imaging techniques are employed to improve the quality of the data, enabling the AI models for the analysis of the sonar images. Extensive analyses are carried out on S3simulator as well as real sonar datasets to validate the performance of AI models for underwater object classification. Our experimental results highlight that the S3Simulator dataset will be a promising benchmark dataset for research on underwater image analysis. https://github.com/bashakamal/S3Simulator.

Keywords: Sonar imagery · Side Scan Sonar Simulated dataset · Segmentation · SelfCAD · Gazebo · underwater object classification

1 Introduction

SONAR, which stands for Sound Navigation and Ranging, plays a crucial role in various underwater applications [20]. Sonar systems utilize sound waves to

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_15.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 219–235, 2025.
https://doi.org/10.1007/978-3-031-78444-6_15

overcome the limitations posed by optical devices, such as water darkness and turbidity. It has found applications in various civilian and defence sectors. The detection and classification of underwater objects in sonar images remain one of the most challenging tasks in marine applications, such as underwater rescue operations, seabed mapping, and coastal management [20].

Traditionally, Sonar imagery is manually inspected by human operators, which is a time-consuming task as well as requires domain expertise [17]. To automate this process, the integration of Artificial Intelligence (AI) emerged as a promising solution. However, the availability of publicly accessible, high-quality sonar datasets to train the AI models is scarce. This paucity of sonar datasets is mainly due to the extensive costs, domain expertise to label, limited resources, security and data sensitivity, and confidentiality constraints. Furthermore, the quality of the available sonar datasets is also suboptimal due to the complexity of the underwater environment, such as various kinds of distortion, underwater noise, speckle noise, small objects, and poor visibility [26].

In order to address the aforementioned challenges, i.e., the scarcity of publicly available sonar data and low-quality sonar images, we propose a synthetic approach for generating **S**ide **S**can **S**onar **s**imulator dataset named as **"S3Simulator"** dataset. A novel framework that combines an advanced AI segmentation model, i.e., Segment Anything Model (SAM) [11], with the self-CAD computer-aided design tool and the 3D dynamic simulator Gazebo is leveraged for the creation of the S3Simulator dataset. Further, it is augmented with cutting-edge computational imaging techniques to provide a heterogeneous dataset replicating real-world sonar imagery such as highlight, shadow, seafloor reverberation, and other characteristics. The S3Simulator dataset consists of 600 images of ships and 600 images of planes, which have been meticulously simulated to replicate real-world sonar conditions.

The proposed S3Simulator dataset is developed in five stages: First, silhouette images of military and civilian planes and ships are acquired in their raw format. In the second stage, a benchmarking segmentation model SAM is utilized to explicitly segment out the image object, i.e., shipwrecks, plane wreckage, and its fragments, from the rest of the image based on the provided prompts. In the third stage, we employ a self-CAD tool to reconstruct 2D segmented images into 3D models, adjusting the model's properties such as shape, size, and texture. In the fourth stage, these 3D objects are deployed on a simulator platform, e.g., Gazebo, to generate a simulated replica of the real-world objects by rendering the self-CAD results. This environment simulates complex sonar characteristics such as noise, shadows, object complexity, and diverse seabed terrain. Finally, we employ a range of computational imaging techniques, including pixel value clipping, linear gradient integration, and nadir zone mask generation, to enhance the data quality and replicate the characteristics of sonar images.

Extensive analysis is carried out on real and S3Simulator datasets to validate the performance of the AI model for underwater image analysis. In particular, we investigate the application of AI models for sonar image classification. To this end, benchmarking classical Machine Learning (ML) approaches such as Support

Vector Machine (SVM), Random forest, K-Nearest Neighbors (KNN) and Deep Learning (DL) models including VGG16, VGG19, MobileNetV2, InceptionResNetV2, InceptionV3, ResNet50, and DenseNet121 are trained using augmentation and transfer learning techniques and tested on unseen real data. The key contributions of the paper are as follows:

- Proposal of a novel **'S3Simulator dataset'** that consists of simulated side-scan sonar imagery to tackle the scarcity of publicly available sonar data and low-quality sonar images.
- Integration of Gazebo simulator and selfCAD 3D with the advanced AI segmentation model SAM, augmented by computational imaging refining.
- Incorporation of a realistic environment comprising images with nadir zone, shadows and object rendering, alongside diverse seabed compositions.
- Extensive evaluation of AI models through classical ML and DL methodologies for sonar image classification in both real-world and simulated scenarios.

The rest of the paper is organized as follows. The related works are described in Sect. 2. The overall architecture of the proposed S3Simulator multi-stage approach and methodology is presented in Sect. 3. In Sect. 4 and Sect. 5, sonar image classification on the S3Simulator dataset and the experimental setup are discussed. In Sect. 6, experimental results are discussed in detail. Finally, the conclusion and future works are enumerated in Sect. 7.

2 Related Works

2.1 Sonar Image Dataset

In the exploration of marine and object detection, researchers have made notable progress in creating sonar image datasets. In one of the earliest studies of side-scan sonar datasets, Huo et al. [9] developed Seabed Objects-KLSG, a side-scan sonar dataset obtained from real sonar equipment, featuring 385 wrecks and 36 drowning victims, 62 airplanes, 129 mines, and 578 seafloor images. Sethuraman et al. [18] AI4Shipwrecks dataset comprises 286 high-resolution side-scan sonar images obtained from autonomous underwater vehicles (AUVs) and labeled with consultation from specialist marine archaeologists. Another dataset i.e. Sonar Common Target Detection Dataset (SCTD) [27] consists of 57 images of planes, 266 images of shipwrecks, and 34 images of drowning victims, each with different dimensions.

However, due to real-world dataset limitations, synthetic sonar images play a significant role in advancing research in underwater exploration. Shin et al. [19] proposed a method using the Unreal Engine (UE) to generate synthetic sonar images with various seabed conditions and objects like cubes, cylinders, and spheres. Sung et al. [21] synthesized realistic sonar images using ray tracing algorithms and GAN. Liu et al. [15] proposed cycle GAN-based generation of realistic acoustic datasets for forward-looking sonars. Yang et al. [25] proposed a side-scan sonar image synthesis method based on the diffusion model. Xi et al. [23] used optical data to train their developed sonar-style image. Lee et al. [13] simulated a realistic sonar image of divers by applying the StyleBankNet image synthesizing scheme to the images captured by an underwater simulator.

2.2 Sonar Image Classification

After the extensive development of sonar imaging technology, underwater image classification has emerged as a crucial area in the field of ocean development. Li et al. [14] used Support Vector Machine (SVM) as the classifier to recognize small diver from dim special diver targets accurately and selected five main characteristics of divers such as divers average scale, velocity, shape, direction, and angle with 94.5% as accuracy rate. After feature extraction, Karine et al. [10] implemented the k-nearest neighbor (KNN) and SVM algorithms for seafloor image classification recorded by side scan sonar. Zhu et al. [28] proposed an extreme learning machine (KELM) and principle component analysis (PCA) for side scan sonar image classification. Du et al. [7] compared different CNN model prediction accuracy and found less improvement for AlexNet and VGG-16 and good improvement for Google Net and ResNet101 after the transfer learning technique is applied. Google Net has the highest prediction accuracy at 94.27%. After fine-tuning limited data, [5] used a pre-trained deep neural network in which ResNet-34 and DenseNet-121 were the best-performing models of underwater image classification.

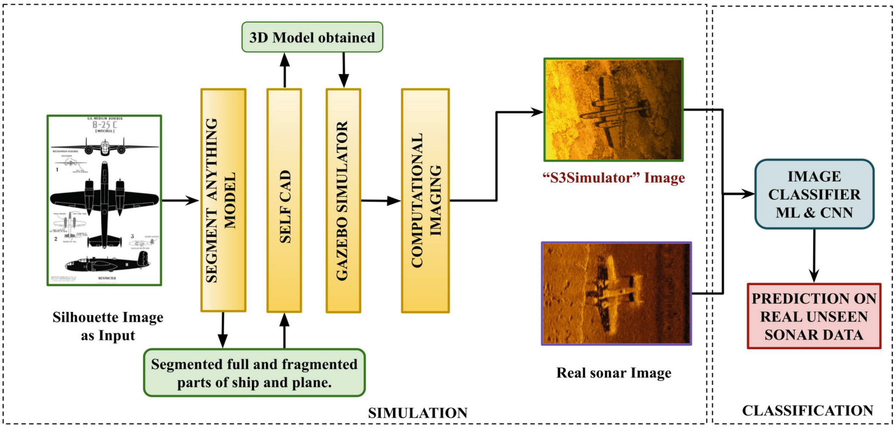


Fig. 1. Overall architecture of proposed S3Simulator-based Sonar Image classification

In contrast to the aforementioned synthetic sonar images, which are expensive and time-consuming due to recreating from real data, our S3Simulator dataset is an economical and time-efficient solution built on simulator technology and advanced AI techniques. To the best of our knowledge, the S3Simulator dataset is the first publicly accessible and extensive compilation of side-scan sonar images for ship and plane objects.

3 S3Simulator Dataset

This section explains the workflow and generation of the S3Simulator dataset. The overall architecture of the proposed pipeline of S3Simulator is depicted in Fig. 1. It consists of modules Segment Anything Model (SAM), SelfCAD, Gazebo, computational imaging, output of the simulated image, real sonar image, and its classification using Machine Learning (ML) and Deep Learning (DL) techniques. The details of the modules are explained below.

3.1 Data Acquisition

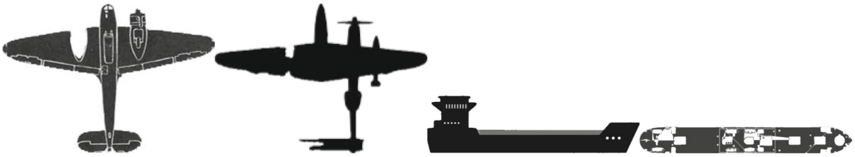
To replicate the sonar imagery of realistic objects such as ships and planes, the data is collected from the Royal Observer Corps Club's third-grade exam. The collection has a total of 62 unique aircraft, whereas each aircraft is depicted as a black image on both sides and plan perspectives [3], and the U.S. ship silhouettes show the relative size of the various classes of aircraft carriers, battleships, cruisers, and destroyers [4]. (The images are represented in the supplementary material for reference.)

Further, for the AI investigation and classification, as mentioned in Sect. 2.1 Seabed object KLSG dataset is utilized. (Sample images of the Seabed object KLSG dataset are given in the supplementary material for reference.) This dataset serves as the basis for testing the S3Simulated dataset against real sonar data.

3.2 Segmentation with Segment Anything Model (SAM)

Segment Anything Model (SAM) [11]- is one of the cutting-edge models in semantic segmentation. SAM is intended to identify and isolate an object of interest within an image in response to specific user-provided prompts. Prompts can be text, a bounding box, a collection of points (including a complete mask), or a single point. Even though the request is ambiguous, the model still generates an appropriate segmentation mask, as shown in Fig. 2. Consequently, it can perform effectively in the zero-shot learning regime, i.e. it can segment objects of types it has never encountered before without the need for further training. SAM consists of an image encoder, a flexible prompt encoder, and a fast mask decoder based on Transformer vision models. The image encoder is applied once per image before prompting the model. Masks consist of dense prompts encoded with convolutions and combined element-wise with the image embedding. Image, prompt, and output token embedding are efficiently mapped to masks via the mask decoder.

In the analysis of real-world objects, SAM is applied to facilitate the segmentation and masking of specified objects. In this work, SAM approach is used to segment the planes and ships objects from the raw silhouette images, as shown in Fig. 2.

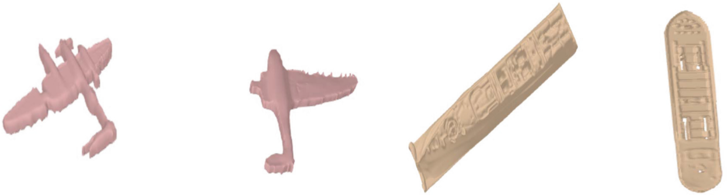


(a) Plane wreck 1 (b) Plane wreck 2 (c) Shipwreck 1 (d) Shipwreck 2

Fig. 2. The Segment Anything Model is utilized to segment fragmented images of ships, aircraft, and vessels from the image.

3.3 3D Model Generation in SelfCAD

SelfCAD [2] is a software application for computer-aided design (CAD) that enables users to modify pre-existing designs as well as to generate 3D model from 2D image. SelfCAD enables users, with its robust tools, to effortlessly create, sculpt, and slice objects. In our work, SelfCAD is employed to generate a 3D model from the segmented 2D images shown in Fig. 2. We refined the 3D models by applying sculpting techniques, improving resolution, modifying tolerances, and manipulating size and shape. The purpose of these modifications is to improve and optimize the final 3D models, which are similar to real-world objects as shown in Fig. 3.

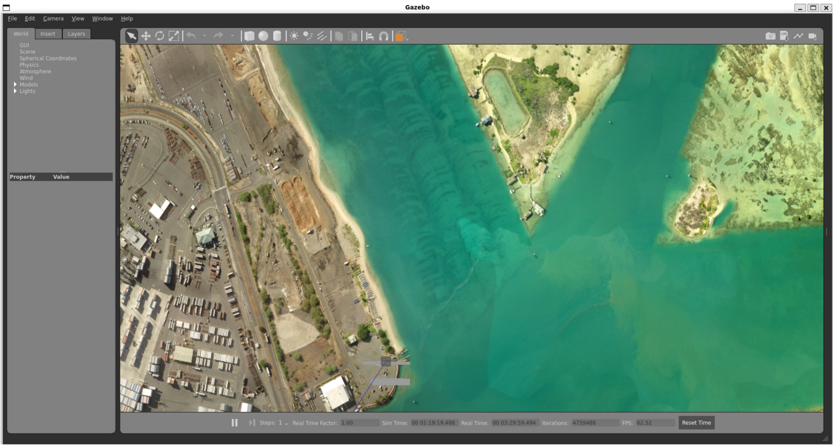


(a) Plane wreck 1 (b) Plane wreck 2 (c) Shipwreck 1 (d) Shipwreck 2

Fig. 3. SelfCAD 3D objects after segmentation

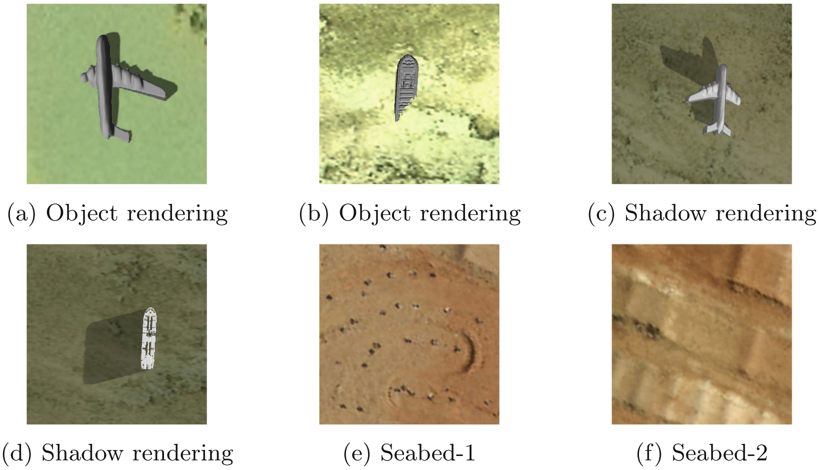
3.4 Deployment to the Gazebo Simulator

Gazebo [1] is an open-source robotic simulator that simplifies high-performance application development. Its primary users are robot designers, developers, and educators. In our work, Gazebo is employed to simulate sonar images by rendering 3D objects and shadows on various seabeds shown in Fig. 5. The generated 3D model is integrated into Gazebo World. The rendering of objects is achieved by adjusting their poses on the x, y, and z axes and incorporating features like roll, pitch, and yaw rotations. Additionally, the visual texture of the 3D model can be fine-tuned with RGB values from the link inspector available in the Gazebo. To bring the simulated image to a more realistic sonar image, we explicitly add



Steps: 1 Real Time Factor: 1.00 Sim Time: 00 01:19:19.488 Real Time: 00 03:29:59.494 Iterations: 4759488 FPS: 62.52

Fig. 4. Gazebo environment



(a) Object rendering

(b) Object rendering

(c) Shadow rendering

(d) Shadow rendering

(e) Seabed-1

(f) Seabed-2

Fig. 5. 3D object simulated in Gazebo with object rendering (a) and (b), shadow rendering (c) and (d) of plane and ship in different seabed (e) and (f).

noise from sensors provided by Gazebo models, which adds Gaussian-sampled disturbance independently to each pixel (Fig. 4).

3.5 Computational Imaging

This process includes a series of computational imaging techniques aimed at converting data from a Gazebo simulator into a visual representation that closely resembles sonar imagery. Some of the key techniques include the clipping of pixel values, the integration of linear gradients, and the generation of nadir zones.

Image Clipping and Integration of Linear Gradient. In a sonar image, we can identify that dark colours represent deeper areas and bright colours represent shallow areas. To mimic the real-world conditions, the linear gradient technique is employed in simulated sonar images by partitioning the image into 50% and applying a gradient on both sides as shown in Fig. 6. The gradient for the image function is given by:

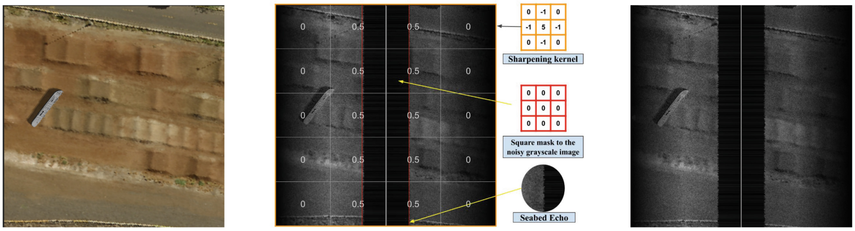
$$\Delta I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right] \tag{1}$$

Gradient for Quarter-based Intensity Mapping in a Simulated Sonar Image,

$$\Delta I(x,y) = \begin{cases} 0 & \text{if } x \leq 0.25 \cdot w \text{ (first quarter)} \\ 0.5 & \text{if } 0.25 \cdot w < x \leq 0.5 \cdot w \text{ (second quarter)} \\ 0.9 & \text{if } 0.5 \cdot w < x \leq 0.75 \cdot w \text{ (third quarter)} \\ 0.5 & \text{if } 0.75 \cdot w < x \leq w \text{ (last quarter)} \end{cases} \tag{2}$$

In this representation:

- w - width of the image.
- $\Delta I(x,y)$ - gradient intensity at position (x,y) in the image.
- The gradient changes at different rates depending on the value of x, where x is the horizontal position within the image.
- The gradient is 0 in the first quarter of the width, 0.5 in the second, 0.9 in the third, and 0 in the last.



(a) Simulated Image (b) Nadir Zone (c) Output

Fig. 6. Image (a) represents the simulated image from the Gazebo; (b) represents the integration of the linear gradient and nadir zone; and (c) represents the final output image after the computational imaging technique.

Generation of Nadir Zone. The nadir zone in the sonar image is beneath the sonar sensor which appears as a dark zone with a thick white line in between the zone. To mimic this in our work the combination of clipping and linear gradient is employed, to mask the nadir zone and thin white line inside the zone.

$$K(x,y) = \begin{cases} 1 & \text{if } I(x,y) \geq Th \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where:

- Th - chosen threshold value.
- $K(x, y)$ - resulting mask, with a value of 1 indicating that the pixel (x, y) is part of the Nidar zone and 0 indicating it's not.

3.6 Image Augmentation

Image augmentation techniques [24] are utilized to improve the resilience and classification performance of the models, trained on both the synthetic sonar simulation dataset and the real sonar dataset. The augmentation pipeline consists of modifications to horizontal flips and random crops. These changes facilitate the inclusion of alterations in the images, hence strengthening the model's ability to acquire robust features and mitigating the issue of overfitting.

4 Sonar Image Classification on S3Simulator Dataset

To showcase the effectiveness of the proposed S3Simulator, the developed dataset is used to benchmark computer vision applications such as image classification. Image classification is a fundamental task in computer vision that involves categorizing an image into one or more predefined classes [12]. This study explores two primary techniques for image classification: the classical Machine Learning (ML) approach, which utilizes algorithms like k-nearest neighbors (KNN), Random Forest, and Support Vector Machines (SVM), suitable for smaller datasets; and deep learning (DL) models, such as Convolutional Neural Networks (CNNs), which can automatically learn from huge datasets with intricate patterns to provide better results.

The k-nearest neighbors [8] algorithm is a simple approach that classifies new data by calculating the distance between the nearest neighbors. Similarly, Random forest [6] is a technique in ensemble learning that combines predictions from multiple decision trees that were trained on random subsets of data. Whereas, SVM [14] utilize a hyperplane in the feature space to distinguish between classes. Formally, the aforementioned techniques can be represented as

$$f(x) = \sum_{i}^{n} \alpha_i y_i K(x, x_i) + b \tag{4}$$

where, $f(x)$ is the predicted label for the input x, α_i is lagrange multipliers obtained from the SVM optimization process, x_i the feature vector of the training data, y_i is class labels (-1 or 1) for the training data point, $K(x, x_i)$ represents the kernel function that calculates the similarity between the input vector x and the support vectors x_i, and b is the bias to determine the offset of the decision boundary.

Deep Learning (DL) is widely used in the field of pattern recognition and are more efficient than traditional machine learning approaches for image classification [16]. In particular, Convolutional Neural Networks (CNN) are utilized to classify images. In our study, we leverage transfer learning approach [22] wherein knowledge from one model is transferred to another model, in order to train deep neural networks with comparatively little data. Mathematically, the neural network learning can be represented as follows.

$$f = \sigma\left(W_n * h + b_n\right) \tag{5}$$

where, f is the predicted label, σ be the activation function, W_n is the weights of the newly added fully-connected layer for binary classification, h is the Output from the pre-trained model (usually the last layer before the final classification layer in the backbone model) and b_n is the biases of the newly added fully-connected layer.

5 Experimental Setup

In this Section, the experimental details employed to develop, train, and evaluate the sonar image classification is explained.

5.1 Dataset

In the generation of the S3Simulator dataset, a 3D object model is generated from the silhouette images. To enhance realism with realistic objects, the silhouette image is acquired from army fighter planes, army bombers, naval planes, and battleships, as mentioned in Sect. 3.1.

For AI investigation, we incorporate a simulated dataset S3Simulator along with a real sonar Seabed objects-KLSG dataset [9]. This dataset comprises 578 seafloor images, 385 wreck images, 36 drowning victim images, 62 aircraft images, and 129 mine images accumulated over a period of more than ten years. With the generous assistance of numerous sonar equipment suppliers—including Lcocean, Hydro-tech Marine, Klein Marine, Tritech, and EdgeTech—this dataset is made possible. Additionally, it comprises images that were obtained directly from the original large-scale side-scan sonar images without any preprocessing.

5.2 Evaluation Protocol

In the evaluation of the sonar image classification task, benchmarking classification metrics such as accuracy and confusion matrix are used. Accuracy is an evaluation metric that allows to measure the total number of predictions a model gets right. Mathematically, Accuracy (ACC) is formulated as,

$$\mathrm{ACC} = \frac{(TP + TN)}{(TP + FP + FN + TN)}, \tag{6}$$

- TP (True Positives) - number of images correctly classified as positive.
- TN (True Negatives) - number of images correctly classified as negative.
- FP (False Positives) - number of images incorrectly classified as positive.
- FN (False Negatives) - number of images incorrectly classified as negative.

Confusion matrix displays counts of the True Positives, False Positives, True Negatives, and False Negatives produced by a model as shown in Fig. 9. Using a confusion matrix we can get the values needed to compute the accuracy of a model.

5.3 Implementation Details

In this investigation, pre-trained models such as VGG16, VGG19, MobileNetV2, InceptionResNetV2, InceptionV3, ResNet50, and DenseNet121 are trained on the ImageNet dataset with two active layers of 1024 and 512 neurons. A dropout layer with a 0.25 dropout rate and a batch normalization layer improved the model's robustness. In training, we used the Adam optimizer with a learning rate of 0.0001 and a batch size of 16. We employed model checkpoint and early stopping during the evaluation to evaluate training progress and prevent overfitting. In the Gazebo, models are described as Simulation Description Format (.sdf) files detailing their physics characteristics, properties, visual appearance, collision, etc. We Utilized Ubuntu 22.04.4 LTS and Gazebo multi-robot simulator, version 11.10.2 for Gazebo simulation. The implementation of image classification is conducted on Google Colab, utilizing an T4 GPU with an allocation of 15 GB of RAM for training and employing Google's TensorFlow framework.

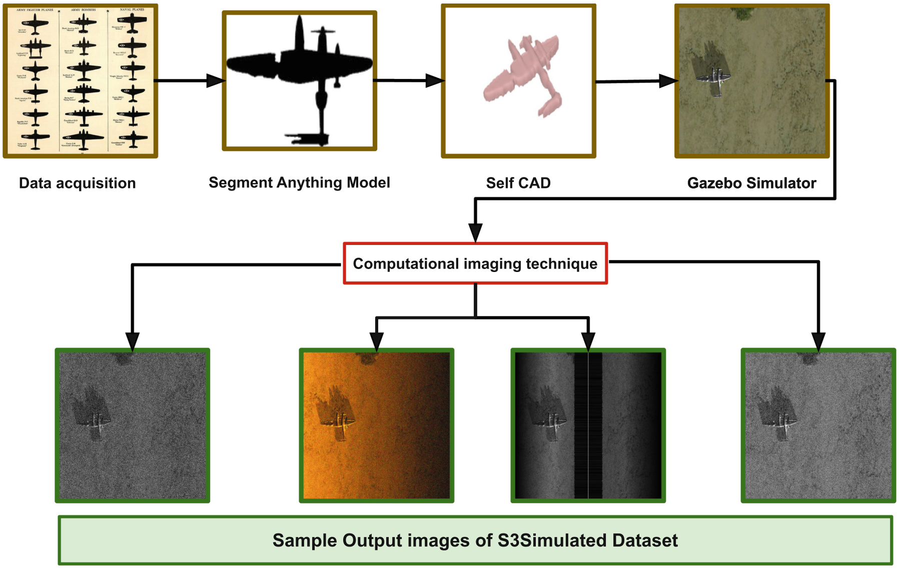

Data acquisition
Segment Anything Model
Self CAD
Gazebo Simulator
Computational imaging technique
Sample Output images of S3Simulated Dataset

Fig. 7. Pipeline of S3Simulated dataset

6 Experimental Results

6.1 S3Simulator Dataset Results

As explained in Sect. 3 overall architecture, the final pipeline of simulated sonar image is shown in Fig. 7. The pipeline consists of data acquisition, SAM, self-CAD, Gazebo simulator, and computational imaging techniques employed to generate the S3Simulator dataset. Sample S3Simulator images shown in Fig. 8.

6.2 Sonar Image Classification

Referring to Sect. 4, a benchmark study of classification using the S3Simulator dataset is carried out using both Machine Learning (ML) and Deep Learning (DL) classifiers. Extensive analysis using simulated data, real data, and real+simulated data is conducted. The classifier performance of ML and DL models are shown in Table 1 and Table 2. From Table 1, it can be observed that by using simulated data SVM outperforms with an accuracy of 92%. Similarly, the Random Forest classifier outperforms with a test accuracy of 88% in real data. While utilizing both real + simulated data and testing on real data, which represents a realistic deployment of scenarios in the wild, SVM classifier outperforms with 88% accuracy. From all the above analyses, SVM classifier was found to be providing superior performance among all.

Analogous to the ML classifier, the performance of the DL classifier, as mentioned in Sect. 5.3, is also investigated. Referring to Table 2, the test accuracy of different models in test data that are trained using real data/real + simulated data are studied. It is observed that while training with real data, DenseNet121 and InceptionResNetV2 outperform the models with an accuracy of 92% and 91%, respectively. Further training with real+simulated data, DenseNet121 achieved the best performance with test accuracy of 96%. It is

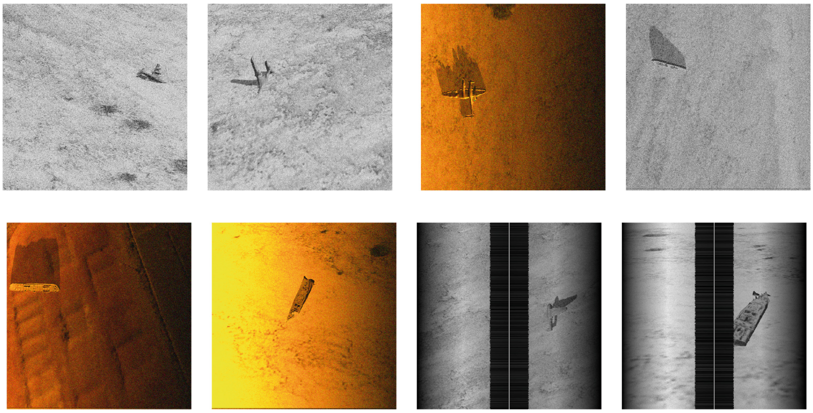


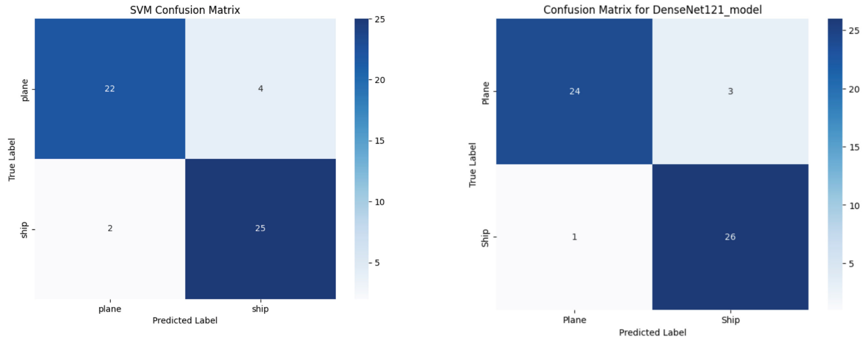
Fig. 8. S3Simulated Dataset images of ship and plane

Table 1. ML Classifier performance of both real and simulated sonar datasets

Training Data	Testing Data	Classifier	Train Accuracy	Test Accuracy
Simulated	Simulated	**SVM**	1.00	**0.92**
		Random Forest	1.00	0.69
		KNN	0.75	0.56
Real	Real	SVM	1.00	0.77
		Random Forest	1.00	**0.83**
		KNN	0.83	0.72
Real + Simulated	Real	**SVM**	1.00	**0.88**
		Random Forest	1.00	0.63
		KNN	0.79	0.65

Table 2. Test accuracy of different models tested in real data

Model	Trained in real data	Trained in real + simulated data	Percentage improved from real data to combined data
VGG_16	0.90	0.94	4%
VGG_19	0.87	0.92	5%
ResNet50	0.64	0.70	6%
InceptionV3	0.91	0.94	3%
DenseNet121	0.92	**0.96**	4%
MobileNetV2	0.89	0.94	5%
InceptionResNetV2	0.91	0.95	4%



(a) ML classifier-SVM

(b) DL Classifier-DenseNet121

Fig. 9. Confusion matrix of the highest-performing classifiers i.e. Support Vector Machine(SVM) and DenseNet121, respectively.

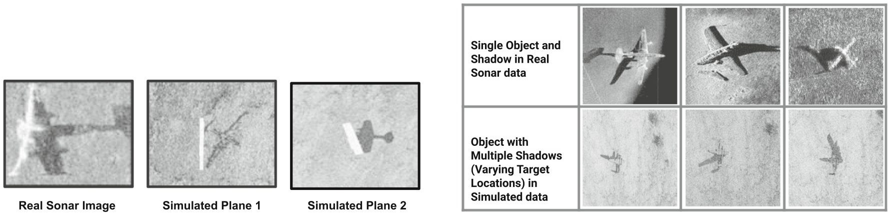
observed that the significant improvement in accuracy of all the models from 3%–6% is observed in real + simulated data compared to real data. This accentuates the impact of additional synthetic data augmenting the training process, by replicating realistic sonar data in terms of the number of images, and quality of images and by recreating real-world scenarios.

The confusion matrices of the best-performed models in both ML (i.e., SVM) and DL (i.e., DenseNet121) are depicted in Fig. 9. The overall as well as class-wise accuracy is analysed in the test scenario. The accuracy of the best ML classifier and DL classifier are 88% and 96%, respectively. It is observed that the performance of the "plane" class is improved in the DL Model with an accuracy of 88% compared to the ML accuracy of 84%. Similarly, the accuracy of "ships" is increased from 92% to 96% in the DL model.

6.3 Visualization: Highlights and Shadows

In sonar image analysis, both the shadow and highlight regions are crucial for accurate classification and detection, as they provide complementary information about the objects' shape and size. In many cases, the highlight area of an object in a sonar image may not be clearly visible, but its shadow can be distinctly observed as shown in Fig. 10(a). By emphasizing the shadow information, the "S3Simulator" dataset addresses an important gap in existing sonar image datasets and provides a valuable resource for researchers to advance the field of sonar image analysis.

The shadow characteristics in publicly available datasets are typically determined by the fixed positioning of the sonar relative to the object. The "S3Simulator" dataset overcomes this limitation by allowing for the generation of sonar images with varying shadow angles for a given object. This is achieved through the flexibility of the simulation process, where the position and orientation of the sonar device can be adjusted to create images with different shadow characteristics, as shown in Fig. 10(b). Furthermore, real-world side-scan sonar data contains a nadir zone, a crucial feature often missing in synthetic datasets. The "S3Simulator" dataset uniquely incorporates the nadir zone, enhancing the realism of simulated sonar imagery as shown in Fig. 11.


Real Sonar Image
Simulated Plane 1
Simulated Plane 2
Single Object and Shadow in Real Sonar data
Object with Multiple Shadows (Varying Target Locations) in Simulated data

(a) Highlighting the Shadow Characteristics in Real and Simulated Sonar Images

(b) Contrasting Singular and Diverse Shadow Representations in Real vs. Simulated Sonar Imagery

Fig. 10. Leveraging Simulated Sonar Shadows to Enhance Real-World Object Identification

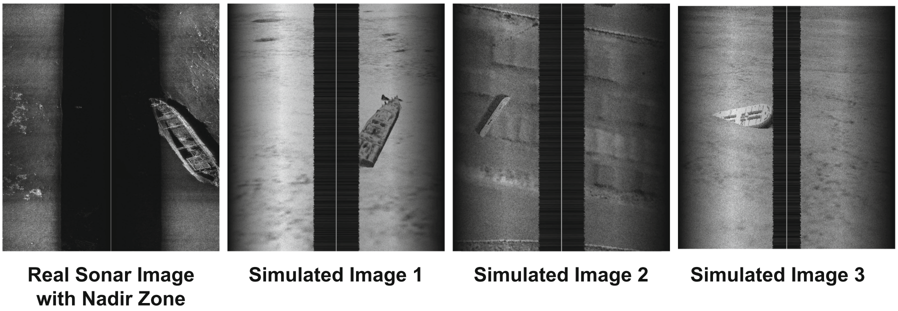


Fig. 11. Addressing the Nadir Zone Challenge in Sonar Simulation through the "S3Simulator" Dataset.

7 Conclusion and Future Works

In this work, we presented a novel benchmarking **S**ide **S**can **S**onar simulator dataset named **"S3Simulator dataset"** for underwater sonar image analysis. By employing a systematic methodology that encompasses the collection of real-world images, reconstruction of 3D models, simulations, and computational imaging techniques, we have effectively generated a comprehensive dataset similar to real-world sonar images. The effectiveness of our methodology is demonstrated by benchmarking image classification results obtained from several machine learning and deep learning techniques applied to both simulated and real sonar datasets. Future enhancements aim to increase reliability and broaden usability by integrating 3D models of humans, mines, and marine life into diverse environmental settings for enhanced information richness. Additionally, we envisage improving the diversity and scalability of the generated datasets by incorporating advanced generative AI techniques such as GAN and diffusion models. We contemplate that the S3Simulator dataset will significantly advance AI technology for marine exploration and surveillance by offering valuable sonar imagery for research purposes.

Acknowledgments. This work was supported by the Naval Research Board (NRB), DRDO, Government of India under grant number: NRB/505/SG/22-23.

References

1. GAZEBO Homepage. https://gazebosim.org/home. Accessed 3 Apr 2024
2. selfCAD Homepage. https://www.selfcad.com/. Accessed 3 Apr 2024
3. Lone_Sentry_Admin: U.S. navy ship silhouettes. https://www.lonesentry.com/blog/u-s-navy-ship-silhouettes.html. Accessed 21 Aug 2024
4. IBCC Digital Archive: Silhouettes of British, American and German aircraft. https://ibccdigitalarchive.lincoln.ac.uk/omeka/collections/document/22384. Accessed 21 Aug 2024

5. Chungath, T.T., Nambiar, A.M., Mittal, A.: Transfer learning and few-shot learning based deep neural network models for underwater sonar image classification with a few samples. IEEE J. Oceanic Eng. **49**(1), 294–310 (2023)
6. Cutler, A., Cutler, D.R., Stevens, J.R.: Random forests. In: Zhang, C., Ma, Y. (eds.) Ensemble Machine Learning, pp. 157–175. Springer, New York (2012). https://doi.org/10.1007/978-1-4419-9326-7_5
7. Du, X., Sun, Y., Song, Y., Sun, H., Yang, L.: A comparative study of different CNN models and transfer learning effect for underwater object classification in side-scan sonar images. Remote Sens. **15**(3), 593 (2023)
8. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
9. Huo, G., Wu, Z., Li, J.: Underwater object classification in sidescan sonar images using deep transfer learning and semisynthetic training data. IEEE Access **8**, 47407–47418 (2020)
10. Karine, A., Lasmar, N., Baussard, A., El Hassouni, M.: Sonar image segmentation based on statistical modeling of wavelet subbands. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–5. IEEE (2015)
11. Kirillov, A., et al.: Segment anything, pp. 4015–4026 (2023)
12. Lai, Y.: A comparison of traditional machine learning and deep learning in image recognition. In: Journal of Physics: Conference Series, vol. 1314, p. 012148. IOP Publishing (2019)
13. Lee, S., Park, B., Kim, A.: Deep learning from shallow dives: sonar image generation and training for underwater object detection. arXiv preprint arXiv:1810.07990 (2018)
14. Li, K., Li, C.L., Zhang, W.: Research of diver sonar image recognition based on support vector machine. Adv. Mater. Res. **785**, 1437–1440 (2013)
15. Liu, D., Wang, Y., Ji, Y., Tsuchiya, H., Yamashita, A., Asama, H.: CycleGAN-based realistic image dataset generation for forward-looking sonar. Adv. Robot. **35**(3–4), 242–254 (2021)
16. O'shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)
17. Rutledge, J., et al.: Intelligent shipwreck search using autonomous underwater vehicles. In: International Conference on Robotics and Automation (ICRA), pp. 6175–6182 (2018)
18. Sethuraman, A.V., et al.: Machine learning for shipwreck segmentation from side scan sonar imagery: dataset and benchmark. arXiv preprint arXiv:2401.14546 (2024)
19. Shin, J., Chang, S., Bays, M.J., Weaver, J., Wettergren, T.A., Ferrari, S.: Synthetic sonar image simulation with various seabed conditions for automatic target recognition. In: OCEANS 2022, Hampton Roads, pp. 1–8. IEEE (2022)
20. Steiniger, Y., Kraus, D., Meisen, T.: Survey on deep learning based computer vision for sonar imagery. Eng. Appl. Artif. Intell. **114**, 105157 (2022)
21. Sung, M., et al.: Realistic sonar image simulation using deep learning for underwater object detection. Int. J. Control Autom. Syst. **18**(3), 523–534 (2020)
22. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big Data **3**, 1–40 (2016)

23. Xi, J., Ye, X.: Sonar image target detection based on simulated stain-like noise and shadow enhancement in optical images under zero-shot learning. J. Mar. Sci. Eng. **12**(2), 352 (2024)
24. Xu, M., Yoon, S., Fuentes, A., Park, D.S.: A comprehensive survey of image augmentation techniques for deep learning. Pattern Recogn. **137**, 109347 (2023)
25. Yang, Z., Zhao, J., Zhang, H., Yu, Y., Huang, C.: A side-scan sonar image synthesis method based on a diffusion model. J. Mar. Sci. Eng. **11**(6), 1103 (2023)
26. Zhang, F., Zhang, W., Cheng, C., Hou, X., Cao, C.: Detection of small objects in side-scan sonar images using an enhanced YOLOv7-based approach. J. Mar. Sci. Eng. **11**(11), 2155 (2023)
27. Zhang, P., Tang, J., Zhong, H., Ning, M., Liu, D., Wu, K.: Self-trained target detection of radar and sonar images using automatic deep learning. IEEE Trans. Geosci. Remote Sens. **60**, 1–14 (2021)
28. Zhu, M., et al.: PCA and kernel-based extreme learning machine for side-scan sonar image classification. In: 2017 IEEE Underwater Technology (UT), pp. 1–4. IEEE (2017)

Trajectory Forecasting Through Low-Rank Adaptation of Discrete Latent Codes

Riccardo Benaglia[1,2](✉), Angelo Porrello[1], Pietro Buzzega[1], Simone Calderara[1], and Rita Cucchiara[1]

[1] AImageLab, University of Modena and Reggio Emilia, Modena, Italy
{angelo.porrello,pietro.buzzega, simone.calderara,rita.cucchiara}@unimore.it
[2] Ammagamma S.r.l., Modena, Italy
riccardo.benaglia@unimore.it

Abstract. Trajectory forecasting is crucial for video surveillance analytics, as it enables the anticipation of future movements for a set of agents, *e.g.*, basketball players engaged in intricate interactions with long-term intentions. Deep generative models offer a natural learning approach for trajectory forecasting, yet they encounter difficulties in achieving an optimal balance between sampling fidelity and diversity. We address this challenge by leveraging Vector Quantized Variational Autoencoders (VQ-VAEs), which utilize a discrete latent space to tackle the issue of posterior collapse. Specifically, we introduce an instance-based codebook that allows tailored latent representations for each example. In a nutshell, the rows of the codebook are dynamically adjusted to reflect contextual information (*i.e.*, past motion patterns extracted from the observed trajectories). In this way, the discretization process gains flexibility, leading to improved reconstructions. Notably, instance-level dynamics are injected into the codebook through low-rank updates, which restrict the customization of the codebook to a lower dimension space. The resulting discrete space serves as the basis of the subsequent step, which regards the training of a diffusion-based predictive model. We show that such a two-fold framework, augmented with instance-level discretization, leads to accurate and diverse forecasts, yielding state-of-the-art performance on three established benchmarks.

Keywords: Trajectory forecasting · Vector Quantization

1 Introduction

Trajectory forecasting finds applications in video surveillance [19], multi-object tracking [6,22], behavioural analysis [29], and intrusion detection [34]. The goal

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_16.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 236–251, 2025.
https://doi.org/10.1007/978-3-031-78444-6_16

is to predict the future paths of a set of agents from a few observations of their motion. The prediction can incorporate the interactions between pedestrians [15, 24, 33], or visual attributes of the environment they move within [5].

As multiple plausible paths can be forecast, trajectory prediction reveals an uncertain and multi-modal nature. To achieve this, recent data-driven approaches [12, 16, 23] lean toward a stochastic formulation that places a distribution over the future trajectory, rather than a single estimated path with 100% certainty (*i.e.*, *deterministic* approaches [24]). In doing so, recent stochastic methods take advantage of the latest breakthroughs in deep generative modeling for image generation. For example, [12, 30] resorted to Generative Adversarial Networks, while [16, 36, 45] borrowed ideas from the class of variational methods.

One of the hindrances toward the application of variational approaches is the *posterior collapse* issue: *i.e.*, when the latent variables collapse to the prior becoming uninformative; as a consequence, the decoder learns to ignore them. This translates into a model with undermined generative capabilities, wherein its predictions are distributed on a single path (*e.g.*, the most trivial one) with low uncertainty. A similar tendency (*mode collapse*) has been observed in adversarial networks, and has been addressed through burdensome learning objectives promoting variety [12, 30], or by devising multiple generator networks [5].

In the field of image generation, **Vector Quantized Variational Autoencoders** [40] (VQ-VAEs) have proven to mitigate posterior collapse. VQ-VAEs models avoid the hand-crafted Gaussian prior distribution; differently, they build upon a learnable categorical prior, thereby yielding a discrete latent space. The symbols of this space are the keys of a fixed-size dictionary (**codebook**), whose values are learnable latent codes. Thanks to the resulting increased flexibility, VQ-VAEs embody a promising paradigm for trajectory forecasting.

In this respect, our main contribution regards the content of the VQ-VAE codebook. In particular, while the original formulation devises a single codebook shared across all examples, we propose to dynamically adjust its values based on the *context* of each example, leading to an **instance-based** codebook. We refer as *context* to the set of historical information related to each agent, namely the past steps of its trajectory as well as its interactions with nearby agents. In this way, we aim to encourage even more flexibility during the discretization process, as distinct motion patterns can be discretized with varying granularity.

Moreover, we envision the customization of the codebook as an **adaptation** of the shared original VQ-VAE codebook. By doing so, our goal is to strike a balance between per-instance customization and the emergence of cross-instance concepts that are relevant across multiple examples. In practice, we draw inspiration from recent advances in Parameter Efficient Fine Tuning and represent the dynamic adjustments to the codebook as **low-rank updates** of its values (see Fig. 1). We show that such a modeling constraint improves the representation capabilities of the learned latent space, thereby encoding additional information and facilitating the reconstruction task. The traditional subsequent stage in VQ-VAEs involves fitting the distribution on the discrete latent codes. In this respect, we make use of a vector-quantized diffusion model [10] to learn the implicit prior, departing from existing approaches [7,40] that rely on autoregressive priors, which are more susceptible to issues related to error accumulation.

The contributions are *i)* to the best of our knowledge, we are the first leveraging VQ-VAEs in a trajectory generation task; *ii)* we introduce a novel instance-based codebook based on low-rank modeling; *iii)* we achieve SOTA performance on three established benchmarks (Stanford Drone [28], NBA [20] and NFL [41]).

2 Related Work

The traditional approach to trajectory prediction considers solely the past movements of the agent [3]. However, its motion is likely to be influenced by the motions of other agents (*e.g.*, to avoid collisions or to perform coordinate actions). The first approaches took into account social behaviors through hand-crafted relations, energy-based features, or rule-based models [2,26]. In recent years, the focus has shifted towards data-driven approaches [1,12], leveraging deep models to extract social information [15,33]. Others, instead, rely on the attention mechanism, which has proven highly effective at capturing interactions within tokenized data [24]. For example, [15] employs a graph-based attention mechanism to model human interactions, while [24] utilizes a social-temporal attention module to capture temporal relationships between consecutive time steps and interpersonal interactions occurring among agents.

Given the inherent uncertainty and multi-modal characteristics of future trajectories, recent approaches embrace a deep probabilistic framework to model their distribution. S-GAN [12] leverages a conditional Generative Adversarial Network (GAN) [8], while the authors of SoPhie [30] extend GANs to incorporates visual and social interaction components. Other works utilize conditional Variational Autoencoders (VAE) [17] for multimodal pedestrian trajectory prediction, including [16,31,36,45,46]. Trajectron++ [31] employs a VAE and represents agents' trajectories in a graph-structured recurrent neural network, while PECNet [23] integrates VAEs and goal conditioning. However, both GAN and VAE-based methods grapple with collapsing issues in trajectory generation, necessitating burdensome countermeasures [38]. Ultimately, the work by [11] pioneers the utilization of denoising diffusion models [13] within the trajectory prediction framework, marking a significant advancement in this domain.

Vector Quantization Models. Vector Quantized Variational Autoencoders [40] address posterior collapse by replacing the continuous latent space of VAEs with a discrete set of codewords. Starting from pioneering works, which showed the potential of these models in image generation [27,40], recent studies focused on improving the two fundamental stages: the codebook learning and the discrete prior learning [18]. In this respect, SQ-VAE [35] replaces deterministic quantization with a pair of stochastic dequantization and quantization processes. To create a more comprehensive codebook, [7] supplements the original training losses of VQ-VAE with adversarial training. Additionally, [47] adopts a masking strategy during training and introduces prior distribution regularization to mitigate issues related to low-codebook utilization.

The advances regarding discrete prior learning involve architectural modifications [7] and a critical reevaluation of autoregression. [37] employs a discrete diffusion architecture to model code prediction, while MaskGIT [4] utilizes a bidirectional transformer decoder. This decoder generates all tokens of an image simultaneously and iteratively refines the image based on the preceding generation. In this paper, we condition the codebook on historical instance-level information while preserving the discrete nature of the latent space.

3 Preliminaries

We denote the future trajectory as $y \in \mathbb{R}^{T \times d}$, where T is the number of future time steps and d is the input channel dimension. When dealing with pedestrians, their trajectories are projected into the 2D bird's-eye view (so $d = 2$). The predicted trajectory $\hat{y}$ is generated by a learnable model, fed with a set of conditioning information: *i)* the observed trajectory $x \in \mathbb{R}^{T_p \times d}$ of the agent, *i.e.*, the coordinates observed at previous T_p steps, and *ii)* a set of neighboring trajectories denoted as $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$. We define neighbors of an agent as all agents within the same scene, without imposing any distance threshold.

Vector Quantization. Standard VAEs [17] employ *i)* an **encoder** $\mathrm{E} \equiv \mathrm{E}(y|\theta_{\mathrm{E}})$ that, given input y, outputs a parametric posterior distribution $q(z|y)$ over latent variable z; *ii)* a **decoder** $\mathrm{G} \equiv \mathrm{G}(z|\theta_{\mathrm{G}})$ that provides the reconstruction of the input data as $p_{\theta_{\mathrm{G}}}(y|z)$. The posterior $q(z|y)$ is encouraged to conform to a standard Gaussian prior distribution $p(z)$, which could lead to over-regularized representations (posterior collapse). VQ-VAEs [40] extend VAEs by employing discrete latent variables and Vector Quantization (VQ) [9]. In particular, both posterior and prior distributions are categorical, and their samples provide indices for a learned **embedding table** $e \in \mathbb{R}^{C \times D}$, which consists of C static D-dimensional latent vectors. As outlined in the following paragraphs, the training of VQ-VAEs is divided into *learning the codebook* and *fitting the categorical prior.*

First Stage. Given the input $y \in \mathbb{R}^{T \times d}$, the encoder provides a continuous representation $z \in \mathbb{R}^{T \times D}$, where $z_t \in \mathbb{R}^D$ with $t \in \{1, 2, \dots, T\}$ and D indicates the dimension of the latent space. Then, the VQ-VAE characterizes the posterior as a joint distribution over T independent **categorical** variables $q(c_1, c_2, \dots, c_T|y)$ (one for each latent). Each marginal $q(c_t|y)$ is determined by matching each element of the encoding sequence z_t with the **nearest** vector in the codebook e:

$$q(c_t|y) = \underbrace{\mathcal{C}(p_1, p_2, \dots, p_C)}_{[0,\dots,0,1,0,\dots,0]} \text{ s.t. } p_c = \begin{cases} 1 & \text{if } c = \mathrm{argmin}_{c' \in \{1,2,\dots,C\}} \|z_t - e_{c'}\|_2^2 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Notably, the posterior distribution is *deterministic* and not stochastic as for VAEs: hence, we can *draw* a sample $z^q \equiv z^q(y)$ from the posterior distribution

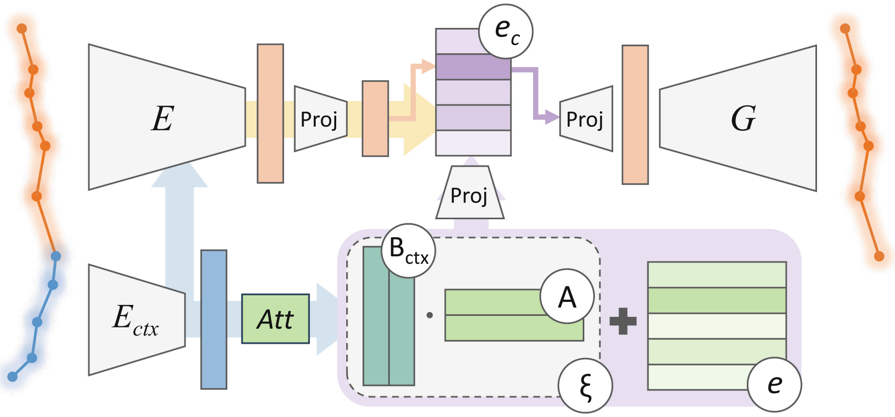


Fig. 1. Overview of our approach to trajectory prediction, based on Vector Quantization and Low-Rank adaptation of the codebook (highlighted in the purple box). (Color figure online)

by **selecting** the corresponding rows of the codebook, as follows:

$$\begin{aligned} z^q &= [e_{c_1}, e_{c_2}, \dots, e_{c_T}] \\ c_t \sim q(c_t|y) &\implies c_t = \operatorname{argmax} q(c_t|y). \end{aligned} \tag{2}$$

The subsequent step regards the decoder G, which reconstructs $\hat{y}$ from the sampled latent vector. During training, the first stage optimizes the following loss:

$$\mathcal{L}_{\text{FS}} = \underbrace{\log p_{\theta_G}(y|z^q)}_{\text{rec. error } e.g., \text{ MSE}} + \sum\nolimits_t \underbrace{\|\texttt{sg}[z_t] - e_{c_t}\|^2}_{\text{embedding loss}} + \sum\nolimits_t \underbrace{\|z_t - \texttt{sg}[e_{c_t}])\|^2}_{\text{commitment loss}}, \tag{3}$$

where `sg` is a shortcut for the `stopgradient` operator, which stops backpropagation from that computational node backward. The second term encourages the quantized latent vectors to be as close as possible to the nearest codeword, while the third one encourages the encoder to be *committed* to the chosen codeword.

Second Stage. The goal here is to learn a parametric model $p_{\theta_p}(c_1, c_2, \dots, c_T)$ – termed *categorical prior* – which allows to draw new samples from the latent space. During this phase, the modules of the VQ-VAE are no longer subject to learning. Given the trained encoder, each training example y is embedded into a sequence of indices, built by relating each latent vector to the nearest row of the codebook (as in Eq. 2). On top of that, the generative model targets the generating process $p(c_1, c_2, \dots, c_T)$ of the discrete latent codes, and optimizes the following Maximum Likelihood Estimation (MLE) training objective:

$$\mathcal{L}_{\text{SS}} = \mathbb{E}_{\substack{c_1,\dots,c_T \\ c_t \sim q(c_t|y)}} [-\log p_{\theta_p}(c_1, c_2, \dots, c_T)]. \tag{4}$$

4 Low-Rank Adaptation for VQ-VAE

We herein present our approach to trajectory prediction, which we name LRVQ, depicted in Fig. 1. Briefly, we exploit VQ-VAEs to encode the future trajectory y of a given agent. On top of that, the following main novelties are introduced:

- We extend VQ-VAE to predict a trajectory coherent with the observed historical trend. To do so, we feed **additional contextual** information to the VQ-VAE, conditioning both the prior and the posterior distributions. The contextual information consists of the past observed trajectory x, and a summary of the interactions between the agent and its neighbours. The structure of the resulting quantization model is presented in Sect. 4.1.
- To encourage further **flexibility**, the codebook itself is conditioned on the additional contextual information (see Sect. 4.2). As discussed later, the context is introduced by devising a **low-rank** adjustment to the codebook.
- To avoid the error accumulation and the *unidirectional bias* problem, typical of auto-regressive methods [10], we make use of a **discrete diffusion model** for the generation of the sequence of indices (see Sect. 4.3). We also introduce a **new sampling technique**, based on the k-means clustering algorithm, to produce better and more consistent generations (see Sect. 4.4).

4.1 Trajectory Forecasting with VQ-VAEs

Formally, our VQ-VAE can be summarized as:

$$h_{\text{ctx}} = \mathrm{E}_{\text{ctx}}([x, \mathcal{X}]) \qquad \text{(context encoding)} \tag{5a}$$

$$z^q = \mathrm{E}(y, [\mathcal{Y}, h_{\text{ctx}}]) \qquad \text{(encoding)} \tag{5b}$$

$$\hat{y} = \mathrm{G}(z^q, \mathcal{Z}^q), \qquad \text{(decoding)} \tag{5c}$$

where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}^q$ represent respectively the past, the future, and the latent quantized representation of the nearby agents' trajectories (see Sect. 3). The modules $\mathrm{E}_{\text{ctx}}(\cdot)$, $\mathrm{E}(\cdot)$, $\mathrm{G}(\cdot)$ are three neural networks, each of which exploits social-temporal transformer [24] to account for social-temporal relations.

In particular, a contextual encoder $\mathrm{E}_{\text{ctx}}(\cdot)$ computes hidden features $h_{\text{ctx}} \in \mathbb{R}^{Tp \times D}$ that summarize both the past trend $x \in \mathbb{R}^{T_p \times 2}$ of the trajectory and spatial interactions (Eq. 5a). The function $\mathrm{E}(\cdot)$ plays the role of the VQ-VAE encoder, transforming the future trajectory y into a discrete representation $z^q \in \mathbb{R}^{T \times D}$ (see Eq. 5b). To condition the model on historical information, the encoder is fed also with the hidden contextual information h_{ctx}; in detail, a tailored cross-attention layer is devised to mix future and past information. Finally, in step (5c) we achieve the estimated future trajectory $\hat{y} \in \mathbb{R}^{T \times 2}$ through the decoder $\mathrm{G}(\cdot)$.

As well as traditional VQ-VAEs, we employ Mean Squared Error (MSE) as our reconstruction term between the ground truth and predicted trajectory.

4.2 Instance-Based Codebook

The codebook plays a crucial role in VQ-VAEs and can cause instabilities during optimization. For instance, the uneven utilization of the vectors of the codebook is a factor that may lead to inefficiencies in representation learning. This imbalance often results in certain elements of the codebook being underutilized, while others never match with real-valued embeddings. To mitigate these issues, the authors of [44] resort to reducing the latent-space dimensionality, showing that it leads to a condensed but richer codebook. In practice, before quantization, each vector z is projected from $\mathbb{R}^D$ to a lower-dimension space $D_r \ll D$. In the following, we will refer to this strategy as **static codebook**, to distinguish it from our proposal that instead leverages dynamic cues.

Our idea is to modify the content of the codebook, such that it reflects the motion observed in the past trajectory. The intuition is that different motion styles (*e.g.*, straight *vs.* curvilinear) could prefer distinct latent codes and discretization strategies. On this basis, we exploit again the contextual features h_{ctx} to generate an **instance-based codebook** $\xi = f_\xi(\cdot, h_{\text{ctx}})$, computed through a tailored learnable module f_ξ. The latter shares the same design of the above-described encoding networks and hence builds upon social-temporal transformers [24]. Afterwards, we combine static and instance-based codebooks by means of summation, thus obtaining a **conditioned** codebook e_c:

$$e_c = \texttt{l2\_norm}(e) + \lambda_\xi \texttt{l2\_norm}(\xi) \tag{6}$$

where `l2_norm` indicates the row-wise l2-normalization $^v/_{\|v\|_2}$ and λ_ξ is an hyper-parameter that weighs the sum. We leverage normalizing layers to ensure that the two components contribute almost equally to the final embedding table.

Moreover, the way we define the codebook draws inspiration from the successes of low-rank adaptation [14] for fine-tuning Large Language Models (LLMs). Namely, we opt for a *low-rank characterization* of f_ξ, which means that the instance-driven modifications to the static codebook lie on a lower-dimensional manifold of the parameter space. We hence define the instance-based codebook ξ as a matrix product of two low-rank matrices B_{ctx} and A, as follow:

$$\begin{aligned} B_{\text{ctx}} &= f_\xi(B, h_{\text{ctx}}) \quad \text{where } B, B_{\text{ctx}} \in \mathbb{R}^{D \times r} \\ \xi &= B_{\text{ctx}} A \quad \text{where } A \in \mathbb{R}^{r \times C}. \end{aligned} \tag{7}$$

Considering B as a set of learnable tokens, f_ξ adopts cross attention between the conditioning information h_{ctx} and B to create an instance-based B_{ctx}.

4.3 Diffusion-Based Categorical Prior

As previously mentioned, the second main stage regards the training of the parametric categorical prior $p_{\theta_p}(c|x, \mathcal{X})$ (note that the p_{θ_p} is also conditioned on historical information), where $c = \{c_1, c_2, \ldots, c_T\}$. Notably, the learned prior serves to forecast the future trajectory y at inference time, when the posterior distribution of y is not available. Section 4.4 provides a detailed description of the sampling procedure, while the rest of this section describes the architectural and training aspects of the categorical prior.

We borrow the design of the categorical prior from the framework of Denoising Diffusion Probabilistic Models (DDPMs). In particular, we employ vector-quantized diffusion models [10], as they naturally handle discrete distributions. Notably, the application of DDPMs allows one to learn the categorical prior without the need for autoregressive modeling, as commonly employed in many existing approaches [7,39]. In the context of trajectory prediction, we view the adoption of a non-autoregressive model as an additional strength. On the one hand, auto-regressive methods can leverage the inherent inductive bias of time-series data, where consecutive time steps relate to each other. However, this often results in error accumulation issues and in the so-called *unidirectional bias* [10], which blurs contextual information that flows in a direction not coherent with the chosen auto-regressive order. In the task under consideration, this means that auto-regressive approaches may struggle to leverage cues emerging in later moments of the trajectory, as *the goal* or the long-range intention of the agent. These crucial aspects of trajectory prediction [23] could be better addressed by the approach proposed in this work, which is **order-free** and capable of capturing multiple plausible trends.

Formally, we define q^{diff} as the diffusion process that injects incremental noise to the token sequence c for Ψ diffusion steps. Instead, p_θ^{diff} is the denoising process that gradually reduces the noise of the noised sequence. The parameters θ of the denoising module are trained with the variational lower bound [32]:

$$\mathcal{L}_{\text{vlb}} = \mathcal{L}^0 + \mathcal{L}^1 + \cdots + \mathcal{L}^{\Psi-1} + \mathcal{L}^{\Psi}, \tag{8a}$$

$$\mathcal{L}^{\psi} = D_{KL}(q^{diff}(c^{\psi}|c^{\psi-1}) \parallel p_\theta^{diff}(c^{\psi}|c^{\psi+1}, \widehat{\mathcal{C}}^{\psi}, x, \mathcal{X})), \tag{8b}$$

$$\mathcal{L}^{c^0} = -\log p_\theta^{diff}(c^0|c^{\psi}, \widehat{\mathcal{C}}^{\psi}, x, \mathcal{X}), \tag{8c}$$

where we use x, $\mathcal{X}$ and $\widehat{\mathcal{C}}^{\psi}$ – the token sequence of neighboring agents at diffusion step ψ – as conditioning information during denoising. (8c) is an auxiliary objective encouraging the prediction of a noiseless token s_0. The loss function:

$$\mathcal{L} \leftarrow \begin{cases} \mathcal{L}^0, & \text{if } \psi = 1 \\ \mathcal{L}^{\psi-1} + \lambda \mathcal{L}^{c^0} & \text{otherwise.} \end{cases} \tag{9}$$

We refer to [10] for more exhaustive details on the diffusion steps and the prior.

Generation. At inference time, the past and social information is encoded using E_{ctx} and then passed to the diffusion process p_θ^{diff}. The latter, after Ψ denoising steps, provides a (denoised) sequence of T indices $\hat{c} \in \mathbb{R}^T$. These indices represent the encoding of the future unobserved trajectory; therefore, we used them to select the proper elements of the codebook e_c, thus allowing us to create a quantized sequence representation z^q. Then z^q undergoes decoding through the VQ-VAE decoder G, which finally yields the generation of trajectories $\hat{y}$.

4.4 Enforcing Effective Multi-modal Forecasting

The sampling approach described above represents the common way to draw new samples from the learned prior of a VQ-VAE. However, we build upon it to create a stronger and richer selection strategy that furthers the multi-modal capabilities of DDPMs. The standard evaluation process involves sampling K distinct trajectories from the model and assessing the top-performing one (as described in Sect. 5). Therefore, each methodology must find the right balance between accuracy in its prediction and potential for exploration. The proposed procedure goes in this direction: we generate numerous *raw* future paths, called *guesses*, and then condense them into the most representative ones. In formal terms, we sample N guesses and then perform the k-means clustering algorithm, with a number of clusters equal to $K < N$ (in our experiments, we set $N = 200$ and $K = 20$). We view the resulting *centroids* as the principal modes of the predictive distribution learned by the DDPM and thus use them for prediction in place of the original samples. This strategy guarantees a twofold advantage compared to naive prediction: firstly, out-of-distribution samples typically form independent clusters, thus enhancing exploration; secondly, the use of centroids reduces the quantization noise, as in-distribution samples are grouped into large clusters and averaged element-wise (see Fig. 2).

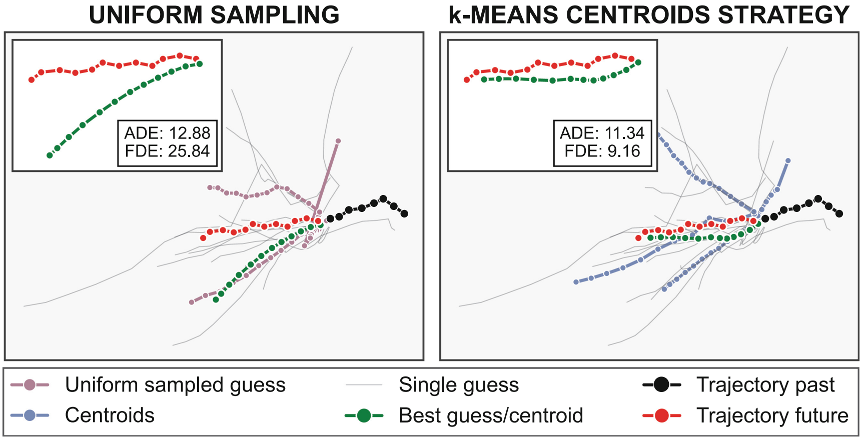


Fig. 2. Comparison between the $K = 5$ samples obtained from a uniform sampling strategy (on the left) and the ones given as output from the proposed k-means centroids sampling strategy (on the right), starting from the same $N = 20$ initial *guesses*.

5 Experiments

We assess our proposal on the following three trajectory prediction benchmarks.

Stanford Drone Dataset (SDD). The dataset [28] gathers trajectories of pedestrians within the Stanford University campus in a bird's eye view. Given 8 time steps (≈3.2 s), methods have to forecast the subsequent 12 frames (4.8 s). We employ the established train-test split [23].

NBA SportVU Dataset (NBA). Collected by the NBA's SportVU automatic tracking system, this dataset [20] provides the trajectories of 10 players and the ball in real basketball games. Given 10 previous time-steps (≈2.0 s), the models predict the subsequent 20 steps (4.0 s).

NFL Football Dataset (NFL). The NFL Football Dataset [41] records the movements of every player throughout each play of the 2017 season. The goal is to predict the trajectories of the 22 players (11 per team) and the ball for the ensuing 3.2 s (16 steps), given the preceding 1.6 s (8 steps).

Metrics. We use two established metrics [1,26] *i.e.*, the Average/Final Displacement Errors (ADE/FDE). Given predicted and ground-truth trajectories, ADE computes the average error on all points, while the FDE restricts the error committed in the final step. Following other works dealing with stochastic models [16,23], we adhere to the best-of-20 protocol [42,43], selecting for evaluation the best trajectory from a pool of $K = 20$ generations. We denote the corresponding metrics as ADE_K and FDE_K; these are in meters for NBA and NFL, and in pixels for SDD. For sports datasets, we compute these metrics at different delta times to provide a more comprehensive assessment.

Table 1. Impact of distinct VQ-VAE codebooks on performance (ADE_{20}/FDE_{20}).

Dataset	Static	Full-Rank	Low-Rank
SDD	8.29/13.44	8.07/12.89	**7.86/12.68**
NBA	0.895/1.279	0.894/1.275	**0.893/1.267**
NFL	0.993/1.702	0.993/1.702	**0.982/1.679**

Implementation Details.[1] We set the number of codewords C to 16 for all datasets, while we take the best rank r for each dataset (*e.g.*, 8 for SDD and NBA, 4 for NFL). For the first stage, we use AdamW [21] as optimizer with $\text{lr} = 5 \times 10^{-4}$, $\beta_1 = 0.5$ and $\beta_2 = 0.9$. We train on SDD for 7000 epochs with batch size equal to 256. For NBA and NFL, we instead optimize for 700 epochs (the batch size equals 64). We use a cosine schedule for λ_ξ from an initial value of 0 to a final value of 1. In this way, we can introduce the instance-level codebook gradually during training.

For the second stage, we re-use the same optimizer/batch-size setup, while training for 3000 epochs for SDD, 1000 epochs for NBA, and 700 for NFL. As an augmentation technique, we rotate the trajectories by a random angle, ranging between 0 and θ_{max}. We set θ_{max} to 180° for the first stage, while we find it beneficial to adopt a lower value (5°) for the second stage.

[1] The code is available at https://github.com/aimagelab/LRVQ.

5.1 On the Impact of the Instance-Based Codebook

To assess the merits of our *low-rank* instance-based codebook, we herein empirically compare it with two alternative strategies. On the one hand, we devise a comparison with a *static* codebook (→ standard VQ-VAEs, lacking instance-level conditioning). Secondly, we contrast it with a *full-rank* codebook (which includes instance-level conditioning but lacks low-rank design constraints). To be more precise, the *full-rank* codebook is a baseline approach herein provided, which computes the values of the codebook through a learnable module fed with historical information as input. Unlike the proposed *low-rank* counterpart, the *full-rank* codebook does not adapt a shared static codebook but directly outputs its values. Through such a comparison, we can evaluate the efficacy of constraining the updates to the dictionary within a low-dimensional manifold.

Table 1 presents the related results: as can be observed, the *low-rank* model outperforms both the *static* and *full-rank* variants. In particular, the improvements are remarkable for SDD and NFL and more modest for NBA. Moreover, the presence of instance-level conditioning, common to *full-* and *low-* approaches, proves particularly beneficial for the SDD dataset, as demonstrated by the gap w.r.t. the static codebook (similar evidence emerges for the NBA dataset).

Table 2. Impact of varying the rank of B on the behavior of the model. Optimal performance (ADE_{20}) is achieved by identifying a sweet spot characterized by a low reconstruction error (ADE_{rec}) and a high accuracy in code prediction (Acc).

Dataset	Rank	$\text{ADE}_{\text{rec}} \downarrow$	Acc(%) ↑	$\text{ADE}_{20} \downarrow$
SDD	4	3.41	26.38	7.96
	16	2.97	22.20	8.06
NBA	4	0.207	15.92	0.898
	16	0.164	13.27	0.892
NFL	4	0.227	15.30	0.982
	16	0.177	11.95	0.996

In the second place, we aim to investigate the impact of the rank r, which controls the dimension of the matrix B_{ctx} (*i.e.*, the degree of instance-level cues introduced into the codebook). In particular, we want to measure how the rank r affects: *i)* the reconstruction capabilities of the VQ-VAE decoder (learned during the first stage); *ii)* the generative capabilities of the diffusion model (learned during the second stage). For point *i)*, we exploit the Average Displacement Error (ADE_{rec}) to assess the reconstruction performance. Instead, to characterize the generative capabilities, we resort to the mean accuracy achieved by the diffusion model in predicting codebook indexes, as well as the already mentioned ADE_{20}.

Table 2 presents the results for different ranks r. We observe that a higher reconstruction capability during the initial training stage is associated with increased difficulty in the diffusion task, resulting in lower accuracy. This indicates a correlation between the two phases: achieving optimal results in the first phase does not necessarily yield the best final generation metrics, as it complicates the joint task of trajectory generation (*i.e.*, sampling from the prior and reconstructing through the decoder). Table 2 demonstrates that the most favorable final metrics are achieved by striking a balance between low reconstruction error and good diffusion accuracy.

5.2 Comparison with SOTA Methods

In this section, we compare our model to the following existing approaches:

- Social-GAN [12] relies on a Conditioned GAN, with a module to handle social interactions between agents.
- Trajectron++ [31] exploits VAEs and graph-structured recurrent networks.
- PECNet [23] augments a VAE with *goal-oriented* reasoning.
- LB-EBM [25] targets the prediction of long-range trajectories through a belief vector, which encapsulates the energy distribution in the environment.
- GroupNet [42] is a multiscale hypergraph network that captures both pair- and group-wise interactions at different scales.
- Memo-Net [43] mimics retrospective memory in neuropsychology and predicts intentions by retrieving similar instances from a memory bank.
- MID [11] leverages a diffusion model to progressively reduce indeterminacy within potential future paths.

Table 3. SDD results (ADE$_{20}$/FDE$_{20}$). * represents the reproduced results from open source. Best results in **bold**, second-best underlined.

Time	S-GAN	Trajectron++	PECNet	MemoNet	GroupNet	MID*	**LRVQ**
4.8 s	27.23/41.44	19.30/32.70	9.96/15.88	8.56/**12.66**	9.31/16.11	9.73/15.32	**7.86**/12.68

Table 4. NBA results (ADE$_{20}$/FDE$_{20}$). Best results in **bold**, second-best underlined.

Time	S-GAN	PECNet	Trajectron++	MemoNet	GroupNet	MID	**LRVQ**
1.0 s	0.41/0.62	0.40/0.71	0.30/0.38	0.38/0.56	0.26/0.34	0.28/0.37	**0.19**/**0.29**
2.0 s	0.81/1.32	0.83/1.61	0.59/0.82	0.71/1.14	0.49/0.70	0.51/0.72	**0.41**/**0.63**
3.0 s	1.19/1.94	1.27/2.44	0.85/1.24	1.00/1.57	0.73/1.02	0.71/0.98	**0.64**/**0.96**
4.0 s	1.59/2.41	1.69/2.95	1.15/1.57	1.25/1.47	0.96/1.30	0.96/**1.27**	**0.89**/**1.27**

Table 5. NFL results (ADE$_{20}$/FDE$_{20}$). Best results in **bold**, second-best underlined.

Time	S-GAN	PECNet	Trajectron++	LB-EBM	GroupNet	MID	**LRVQ**
1.0 s	0.37/0.68	0.52/0.97	0.41/0.65	0.75/1.05	0.32/0.57	0.30/0.58	**0.23**/**0.35**
2.0 s	0.83/1.53	1.19/2.47	0.93/1.65	1.26/2.28	0.73/1.39	0.71/1.31	**0.53**/**0.92**
3.2 s	1.44/2.51	1.99/3.84	1.54/2.58	1.90/3.25	1.21/2.15	1.14/1.92	**0.98**/**1.68**

We report the comparison in Table 3, Table 4, and Table 5. To sum up, our LRVQ demonstrates superior performance across all the considered benchmarks.

On the SDD dataset (Table 3), we attain superior ADE results, matching closely MemoNet in FDE. While PECNet and GroupNet, among C-VAE methods, demonstrate noteworthy performance compared to the older S-GAN and Trajectron++, they struggle in FDE, especially when compared to MemoNet. This could be ascribed to the effective sampling strategy of MemoNet, which integrates a tailored clustering phase to generate multiple overall intentions.

Additionally, our approach showcases robust performance across all examined partial timestamps for both the NBA (Table 4) and NFL datasets (Table 5). The two most competing methods are GroupNet – based on the C-VAE framework – and more importantly MID, which akin to our approach utilizes a diffusion process. However, we highlight an important distinction with MID, which we consider as a motivation for our improvements: while MID adopts diffusion modeling directly in output space, we instead apply it to the discrete variables extracted by the VQ-VAE encoder. We believe that our latent-based formulation further promotes the emergence of multi-modal generative capabilities.

5.3 Qualitative Results

Figure 3 provides a qualitative comparison on 20 generations (with sub-sampling) produced by a VQ-VAE trained with a *static* codebook, a *dynamic* codebook,

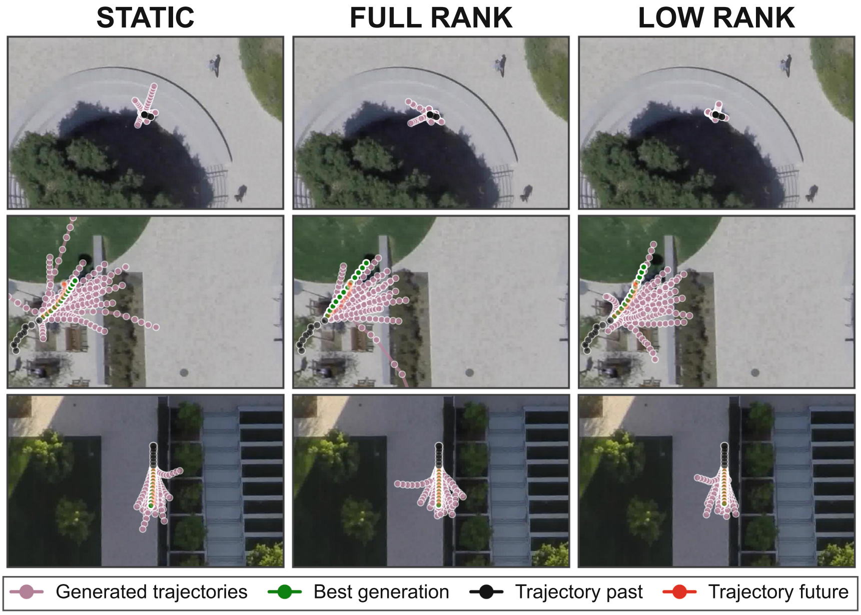


Fig. 3. Qualitative comparison for three SDD scenes (one for each row of the figure) between the trajectories obtained from a VQ-VAE with a static codebook, a full rank codebook the proposed *low-rank* codebook (from left to right).

and the *low-rank* conditioned codebook (see Sect. 5.1). Each row illustrates a different scene from the SDD dataset, showcasing different agent behaviors: in the first one, the agent remains stationary, while in the others, it either turns left or proceeds straight ahead. Compared to the other two methods, low-rank conditioning appears to be more accurate, particularly in complex scenarios where the agent stays still or changes its direction of movement.

6 Discussion and Conclusions

Limitations. The complexity of our model is linked to two factors:

- Two-step training procedure: although VQ-VAE offers benefits such as a learned prior, the training must be divided into two distinct stages, which increases the total time required to train the model.
- Inference time: the inference procedure described in Sect. 4.4 takes longer as the number N of starting guesses increases. To obtain a trade-off between the accuracy of the ensemble of K final generations and the computational time, the parameter N has to be carefully adjusted.

Conclusion. We propose a stochastic approach for trajectory prediction. It builds upon Vector Quantization to yield a predictive distribution that preserves both sampling fidelity and diversity. Our main contribution lies in a dynamic, instance-related codebook encompassing past trajectory information. Notably, contextual information is incorporated into the codebook through a low-rank update. We conduct several empirical studies to validate our approach, demonstrating its superior generative capabilities compared to both standard VQ-VAEs and existing methods. This leads to state-of-the-art results on three established benchmarks.

Acknowledgement. The research was supported by the Italian Ministry for University and Research through the PNRR project ECOSISTER ECS 00000033 CUP E93C22001100001 and by the EU Horizon project "ELIAS - European Lighthouse of AI for Sustainability" (No. 101120237).

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: CVPR (2016)
2. Antonini, G., Bierlaire, M., Weber, M.: Discrete choice models of pedestrian walking behavior. Transp. Res. B Methodol. **40** (2006)
3. Becker, S., Hug, R., Hubner, W., Arens, M.: RED: a simple but effective baseline predictor for the trajnet benchmark. In: ECCVW (2018)
4. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: MaskGIT: masked generative image transformer. In: CVPR (2022)

5. Dendorfer, P., Elflein, S., Leal-Taixé, L.: MG-GAN: a multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In: ICCV (2021)
6. Dendorfer, P., Yugay, V., Osep, A., Leal-Taixé, L.: Quo Vadis: is trajectory forecasting the key towards long-term multi-object tracking? In: Advances in Neural Information Processing Systems 35, pp. 15657–15671 (2022)
7. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
8. Goodfellow, I., et al.: Generative adversarial nets. In: NeurIPS (2014)
9. Gray, R.M., Neuhoff, D.L.: Quantization. IEEE Trans. Inf. Theory **44** (1998)
10. Gu, S., et al.: Vector quantized diffusion model for text-to-image synthesis. In: CVPR (2022)
11. Gu, T., et al.: Stochastic trajectory prediction via motion indeterminacy diffusion. In: CVPR (2022)
12. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: CVPR (2018)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
14. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. In: ICLR (2021)
15. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: STGAT: modeling spatial-temporal interactions for human trajectory prediction. In: ICCV (2019)
16. Ivanovic, B., Pavone, M.: The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: ICCV (2019)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
18. Kolesnikov, A., Susano Pinto, A., Beyer, L., Zhai, X., Harmsen, J., Houlsby, N.: UViM: a unified modeling approach for vision with learned guiding codes. In: NeurIPS (2022)
19. Li, Y., Liang, R., Wei, W., Wang, W., Zhou, J., Li, X.: Temporal pyramid network with spatial-temporal attention for pedestrian trajectory prediction. IEEE TNSE (2021)
20. linouk23: NBA player movements. https://github.com/linouk23/NBA-Player-Movements. Accessed 2016
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
22. Mancusi, G., Panariello, A., Porrello, A., Fabbri, M., Calderara, S., Cucchiara, R.: TrackFlow: multi-object tracking with normalizing flows. In: ICCV (2023)
23. Mangalam, K., et al.: It is not the journey but the destination: endpoint conditioned trajectory prediction. In: ECCV (2020)
24. Monti, A., Porrello, A., Calderara, S., Coscia, P., Ballan, L., Cucchiara, R.: How many observations are enough? Knowledge distillation for trajectory forecasting. In: CVPR (2022)
25. Pang, B., Zhao, T., Xie, X., Wu, Y.N.: Trajectory prediction with latent belief energy-based model. In: CVPR (2021)
26. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: ICCV (2009)
27. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with VQ-VAE-2. In: NeurIPS (2019)
28. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: human trajectory understanding in crowded scenes. In: ECCV (2016)
29. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: a survey. IJRR **39** (2020)

30. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints. In: CVPR (2019)
31. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. In: ECCV (2020)
32. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
33. Sun, C., Karlsson, P., Wu, J., Tenenbaum, J.B., Murphy, K.: Stochastic prediction of multi-agent interactions from partial observations. In: ICLR (2019)
34. Sun, J., Chen, J., Chen, T., Fan, J., He, S.: PIDNet: an efficient network for dynamic pedestrian intrusion detection. In: ACM Multimedia (2020)
35. Takida, Y., et al.: SQ-VAE: variational bayes on discrete representation with self-annealed stochastic quantization. In: ICML (2022)
36. Tang, C., Salakhutdinov, R.R.: Multiple futures prediction. In: NeurIPS (2019)
37. Tang, Z., Gu, S., Bao, J., Chen, D., Wen, F.: Improved vector quantized diffusion models. arXiv preprint (2022)
38. Thiede, L.A., Brahma, P.P.: Analyzing the variety loss in the context of probabilistic trajectory prediction. In: ICCV (2019)
39. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: ICML (2016)
40. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. In: NeurIPS 30 (2017)
41. a vhadgar: Big data bowl. https://github.com/a-vhadgar/Big-Data-Bowl. Accessed 2017
42. Xu, C., Li, M., Ni, Z., Zhang, Y., Chen, S.: GroupNet: multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In: CVPR (2022)
43. Xu, C., Mao, W., Zhang, W., Chen, S.: Remember intentions: retrospective-memory-based trajectory prediction. In: CVPR (2022)
44. Yu, J., et al.: Vector-quantized image modeling with improved VQGAN. In: ICLR (2022)
45. Yuan, Y., Kitani, K.: Diverse trajectory forecasting with determinantal point processes. In: ICLR (2020)
46. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: AgentFormer: agent-aware transformers for socio-temporal multi-agent forecasting. In: ICCV (2021)
47. Zhang, J., Zhan, F., Theobalt, C., Lu, S.: Regularized vector quantization for tokenized image synthesis. In: CVPR (2023)

Towards Completeness: A Generalizable Action Proposal Generator for Zero-Shot Temporal Action Localization

Jia-Run Du[1], Kun-Yu Lin[1], Jingke Meng[1](✉), and Wei-Shi Zheng[1,2,3]

[1] School of Computer Science and Engineering,
Sun Yat-sen University, Guangzhou, China
mengjke@gmail.com
[2] Key Laboratory of Machine Intelligence and Advanced Computing,
Ministry of Education, Guangzhou, China
[3] Guangdong Province Key Laboratory of Information Security Technology,
Sun Yat-sen University, Guangzhou, China

Abstract. To address the zero-shot temporal action localization (ZSTAL) task, existing works develop models that are generalizable to detect and classify actions from unseen categories. They typically develop a category-agnostic action detector and combine it with the Contrastive Language-Image Pre-training (CLIP) model to solve ZSTAL. However, these methods suffer from incomplete action proposals generated for *unseen* categories, since they follow a frame-level prediction paradigm and require hand-crafted post-processing to generate action proposals. To address this problem, in this work, we propose a novel model named Generalizable Action Proposal generator (GAP), which can interface seamlessly with CLIP and generate action proposals in a holistic way. Our GAP is built in a query-based architecture and trained with a proposal-level objective, enabling it to estimate proposal completeness and eliminate the hand-crafted post-processing. Based on this architecture, we propose an Action-aware Discrimination loss to enhance the category-agnostic dynamic information of actions. Besides, we introduce a Static-Dynamic Rectifying module that incorporates the generalizable static information from CLIP to refine the predicted proposals, which improves proposal completeness in a generalizable manner. Our experiments show that our GAP achieves state-of-the-art performance on two challenging ZSTAL benchmarks, *i.e.*, Thumos14 and ActivityNet1.3. Specifically, our model obtains significant performance improvement over previous works on the two benchmarks, *i.e.*, +3.2% and +3.4% average mAP, respectively. The code is available at https://github.com/Run542968/GAP.

Keywords: Zero-Shot Learning · Temporal Action Localization

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_17.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 252–267, 2025.
https://doi.org/10.1007/978-3-031-78444-6_17

1 Introduction

Temporal Action Localization (TAL) is one of the most fundamental tasks in video understanding, which aims to *detect* and *classify* action instances in long untrimmed videos. It is important for real-world applications such as video retrieval [6,21,30,44], anomaly detection [8,40,48], action assessment [18,19], and highlight detection [11,32]. In recent years, many methods have shown significant performance in the close-set setting [7,10,28], where categories are consistent between training and inference. However, a model trained in the close-set setting is capable of localizing only pre-defined action categories. For example, a model trained on a gymnastic dataset cannot localize a "diving" action, even though they are both sports actions. As a result, temporal action localization models are significantly limited in real-world applications.

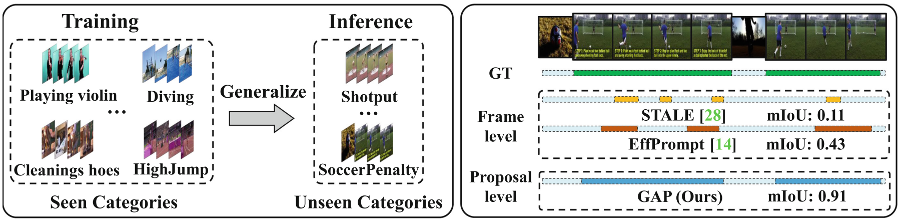


Fig. 1. Left: Zero-shot temporal action localization requires the model trained on *seen* action categories to be generalizable in detecting and classifying *unseen* action categories during inference. **Right:** Visualization of the action proposals generated by STALE [33], EffPrompt [14] and our GAP. The "mIoU" denotes the mean Intersection over Union, which evaluates the completeness of predicted proposals. We can find that our GAP generates more complete action proposals and has a higher mIoU score than the compared frame-level methods. Best viewed in color.

To alleviate the above limitation, our work studies the Zero-Shot Temporal Action Localization (ZSTAL) task. This task aims to develop a localization model capable of localizing actions from *unseen* categories by training with only *seen* categories. In this task, the action categories in training and inference are disjoint, that is neither labels nor data for testing categories are available during training. For example, as shown in Fig. 1 (Left), ZSTAL aims to develop a model that is capable of localizing instances of "Shotput" by training with instances of "Diving", "HighJump", *etc.*

Typically, existing works address the ZSTAL task by a composable model, which consists of a CLIP-based classifier for action classification and a category-agnostic action detector for detecting instances of unseen action categories. For example, Ju *et al.* [14] propose to combine the Contrastive Language-Image Pre-training (CLIP) model [36] with an off-the-shelf frame-level action detector to solve the ZSTAL task. STALE [33] design a single-stage model that consists of a parallel frame-level detector and CLIP-based classifier for ZSTAL.

Despite the progress made by these methods, they suffer from generating *incomplete proposal* in detecting *unseen* action categories. As shown in Fig. 1 (Right), the frame-level detectors (*i.e.*, STALE [33] and EffPrompt [14]) generate fragmented action proposals and have low mIoU scores when detecting unseen category "SoccerPenalty". This is because these detectors are trained with frame-level objectives and require hand-crafted post-processing (*e.g.*, aggregating frame-level predictions via threshold) to obtain action proposals, which leads to a lack of training on estimating the completeness of action proposals.

In this work, we propose a novel Generalizable Action Proposal generator named GAP, aiming to generate complete proposals of action instances for unseen categories. Our proposed GAP is designed with a query-based architecture, enabling it to estimate the completeness of action proposals through training with proposal-level objectives. The proposal-level paradigm eliminates the need for hand-crafted post-processing, supporting seamless integration with CLIP to address ZSTAL. Based on the architecture, our GAP first models category-agnostic temporal dynamics and incorporates an Action-aware Discrimination loss to enhance dynamic perception by distinguishing actions from background. Furthermore, we propose a novel Static-Dynamic Rectifying module to integrate generalizable static information from CLIP into the proposal generation process. The Static-Dynamic Rectifying module exploits the complementary nature of static and dynamic information in actions to refine the generated proposals, improving the completeness of action proposals in a generalizable manner.

Overall, our main contributions are as follows:

- We propose a novel Generalizable Action Proposal generator named GAP, which can generate action proposals in a holistic way and eliminate the complex hand-crafted post-processing.
- We propose a novel Static-Dynamic Rectifying module, which integrates generalizable static information from CLIP to refine the generated proposals, improving the completeness of action proposals for unseen categories in a generalizable manner.
- Extensive experimental results on two challenging benchmarks, *i.e.*, Thumos14 and ActivityNet1.3, demonstrate the superiority of our method. Our approach significantly improves performance over previous work, +3.2% and +3.4% in terms of average mAP, on the two benchmarks, respectively.

2 Related Works

2.1 Temporal Action Localization

Temporal Action Localization (TAL) is one of the key tasks in video understanding topics. Existing methods can be roughly divided into two categories, namely, two-stage methods and one-stage methods. The one-stage methods [28,38,49] do the detection and classification with a single network. Two-stage [20,25,26,46,47] methods split the localization process into two stages: proposal generation and proposal classification. Most of the previous works put emphasis on the proposal generation phrase [25,26,39,41]. Concretely, boundary-based [20,25,26] predict the probability of the action boundary and densely match the start and end timestamps according to the prediction score. Query-based methods [39,41] directly generate action proposals based on the whole feature sequence and fully leverage the global temporal context. In this work, we employ query-based architecture and focus on integrating generalizable static and dynamic information to improve the completeness of action proposals generated for *unseen categories*.

2.2 Zero-Shot Temporal Action Localization

Zero-shot temporal action localization (ZSTAL) is concerned with the problem of detecting and classifying unseen categories that are not seen during training [14,15,33,35]. This task is of significant importance for real-world applications because the available training data is often insufficient to cover all the action categories in practical use. Recently, EffPrompt [14] is the pioneering work to utilize the image-text pre-trained model CLIP [36] for ZSTAL, which adopts an action detector (*i.e.*, AFSD [25]) for action detection and apply the CLIP for action classification [23,24,43,51,52]. Subsequently, STALE [33] and ZEETAD [35] trains a single-stage model that consists of a parallel frame-level detector and classifier for ZSTAL. Despite the process made by these methods, they struggle to generate complete action proposals for action in unseen categories. In this work, we focus on building a proposal-level action detector, which integrates generalizable static-dynamic information to improve the completeness of action proposals.

2.3 Vision-Language Pre-training

The pre-trained Vision-Language Models (VLMs) have showcased significant potential in learning generic visual representation and enabled zero-shot visual recognition. As a representative work, the Contrastive Language-Image Pre-training (CLIP) [36] was trained on 400 million image-text pairs and showed excellent zero-shot transferable ability on 30 datasets. In the video domain, similar ideas have also been explored for video-text pre-training [3,45] with a large-scale video-text dataset Howto100M [31]. However, due to the videos containing more complex information (*e.g.*, temporal relation) than images and large-scale paired video-text datasets being less available, video-text pre-training still has room for development [5,12,17,22,45,50]. In this work, we develop a generalizable action detector that can seamlessly interface with the CLIP, thus utilizing the excellent zero-shot recognition ability of CLIP to solve the zero-shot temporal action localization problem.

3 Methodology

In this section, we detail our GAP, a novel Generalizable Action Proposal generator that integrates generalizable static-dynamic information to improve the completeness of generated action proposals.

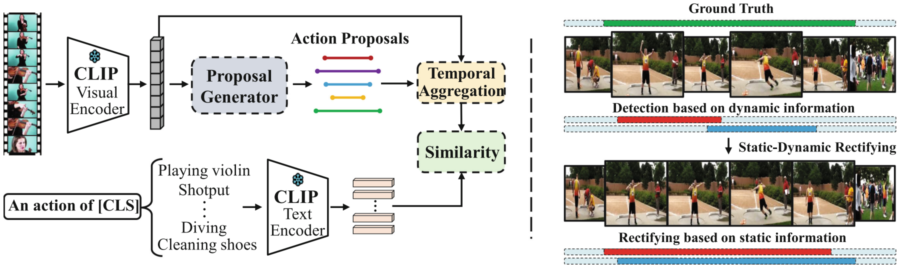


Fig. 2. Left: The pipeline of our method. We adopt a video of $T = 8$ with $N_q = 5$ predicted action proposals for example. **Right:** An illustration of the motivation of Static-Dynamic Rectifying. The red and blue areas in the horizontal bar represent two predicted action proposals. *Top:* Detection by leveraging only dynamic information may result in incomplete proposals, where the model focuses on salient dynamic parts. *Bottom:* After cooperating with static and dynamic information, the proposals are refined by interacting with proposals exhibiting consistent static information to approach ground truth. Best viewed in color.

3.1 Problem Formulation

Zero-Shot Temporal Action Localization (ZSTAL) aims to *detect* and *classify* action instances of unseen categories in an untrimmed video, where the model is trained only with the seen categories. Formally, the category space of ZSTAL is divided into the seen set C^s and unseen set C^u, where $C = C^s \cup C^u$ and $C^s \cap C^u = \varnothing$. Each training video $\mathcal{V}$ is labeled with a set of action annotations $\mathcal{Y}_{gt} = \{t_i, c_i\}_{i=1}^{i=N_{gt}}$, where $t_i = (t_i^s, t_i^e)$ represents the duration (*i.e.*, action proposal) of the action instance, where t_i^s and t_i^e are start and end timestamps, $c_i \in C^s$ is the category and N_{gt} is the number of action instances in video $\mathcal{V}$. In the inference phase, the model needs to predict a set of action instances $\mathcal{Y}_{pre} = \{\tilde{t}_i, \tilde{c}_i\}_{i=1}^{i=N_q}$ that has the same form as $\mathcal{Y}_{gt}$ for each video, where N_q is the number of predicted action proposals in inference, and $\tilde{c}_i \in C^u$.

3.2 Model Overview

Pipeline of Our Method. Our model is composed of a CLIP-based action classifier and an action detector (*i.e.*, proposal generator), as shown in Fig. 2 (Left). The action detector generates category-agnostic action proposals for unseen action categories. Then, the action classification is achieved by utilizing the excellent zero-shot recognition abilities of CLIP, where a temporal aggregation module is adopted to aggregate frame features for similarity computation.

The Proposed Action Detector. The core of our work is the proposal-level action detector GAP, which integrates generalizable static-dynamic information to improve the completeness of generated action proposals. As shown in Fig. 3, the GAP is designed with a query-based architecture for temporal modeling, and an Action-aware Discrimination loss $\mathcal{L}_{ad}$ is used to enhance the perception of category-agnostic temporal dynamics. Then, to mitigate the incomplete problem introduced by category-agnostic modeling, a novel Static-Dynamic Rectifying module is proposed to incorporate static information from CLIP to refine the generated proposals, improving the completeness of action proposals.

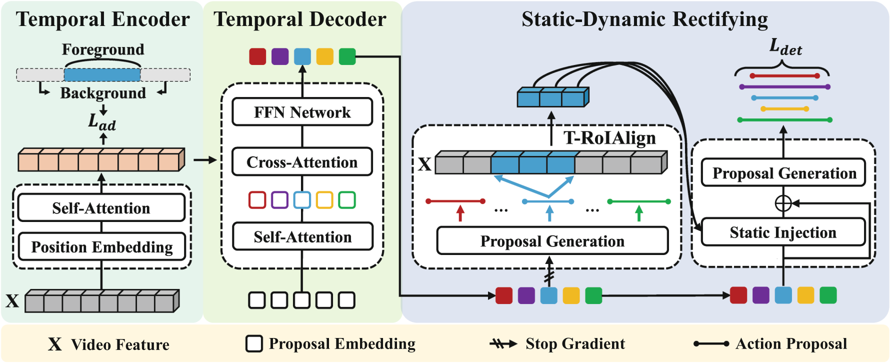


Fig. 3. An illustration of our proposed GAP. Specifically, given the video feature X extracted by the visual encoder, which is fed into the temporal encoder for temporal dynamics modeling. And an Action-aware Discrimination loss $\mathcal{L}_{ad}$ is used to enhance the temporal modeling by distinguishing action from the background. Next, the temporal decoder is adopted to generate dynamic-aware action queries. Then, the static information is injected into dynamic-aware action queries by the Static-Dynamic Rectifying module for refinement. Finally, action proposals are generated and supervised by the detection loss $\mathcal{L}_{det}$. Best viewed in color.

3.3 Temporal Dynamics Modeling

In this section, we design a query-based proposal generator with the transformer [4,42] structure for temporal modeling, which incorporates an Action-aware Discrimination loss to enhance dynamics perception by distinguishing actions from background.

Query-Based Architecture. Following previous works [14,33], we use the visual encoder $\mathcal{F}_v$ of CLIP [36] for video feature extraction. Specifically, the frames of video $\mathcal{V}$ are fed into $\mathcal{F}_v$ to obtain features $X = \mathcal{F}_v(\mathcal{V}) \in \mathbb{R}^{T \times D}$, where

T denotes the number of frames, D is the feature dimension. Subsequently, the video features X are fed into the temporal encoder, where the position embedding and self-attention are applied to model the temporal relation within them. After that, the temporal features $\hat{X} \in \mathbb{R}^{T \times D}$ are obtained.

Given the temporal features $\hat{X}$, they are fed into the temporal decoder along with a set of learnable action queries $\mathcal{Q}$. The action queries $\mathcal{Q} = \{q_i\}_{i=1}^{i=N_q}$, where q_i is learnable vector with random initialization. As shown in Fig. 3, in the decoder, the module follows the order of the self-attention module, cross-attention module, and feedforward network. Specifically, self-attention is adopted among the action queries to model the query relations with each other. The cross-attention performs the interactions between the action queries with the temporal features $\hat{X}$, thereby the action queries can integrate the rich temporal dynamics from video. Finally, the dynamic-aware action queries $\hat{\mathcal{Q}}$ are obtained after the feedforward network.

Temporal Dynamics Enhancement. In order to enhance the temporal feature modeled by the temporal encoder, we propose an Action-aware Discrimination loss $\mathcal{L}_{ad}$ by identifying whether each frame contains an action, which is formulated as follows:

$$\mathcal{L}_{ad} = -\sum_{i=1}^{T}(m_i \log(\sigma(a_i)) + (1 - m_i)\log(1 - \sigma(a_i))), \tag{1}$$

where σ is the sigmoid function, and a_i $(i \in [1, T])$ is the actionness score for i-th frame, which is predicted by feeding temporal features $\hat{X}$ into a 1D convolutional network. m_i is obtained by mapping the action boundary timestamps in ground truth $\mathcal{Y}_{gt}$ to temporal foreground-background mask $\{m_i\}_{i=1}^{i=T}$ as follows:

$$m_i = \begin{cases} 1, \text{ if } \frac{i}{T} \in [t^s, t^e] \\ 0, \text{ if } \frac{i}{T} \notin [t^s, t^e], \end{cases} \tag{2}$$

where $[t^s, t^e] \in \mathcal{Y}_{gt}$ is the normalized [start, end] timestamps of each action instance.

With the Action-aware Discrimination loss $\mathcal{L}_{ad}$, the temporal encoder is capable of perceiving more category-agnostic dynamics of actions, thus helping to generate more complete action proposals for unseen categories.

3.4 Static-Dynamic Rectifying

Since actions are composed of static and dynamic aspects [1], by only using dynamic information of action, the generator tends to predict regions exhibiting salient dynamics, rather than generating complete proposals that are close to the ground truth. For example, as shown in Fig. 2 (Right), the action proposals generated leveraging dynamic information are mainly located in the regions with intense motion in the “Shotput” action, such as “turning” and “bending the elbow”.

Motivated by the above, we propose to integrate generalizable static and dynamic information to improve the completeness of action proposals. We propose a Static-Dynamic Rectifying module, which injects the static information from CLIP into the dynamic-aware action queries $\hat{Q}$. As shown in Fig. 2 (Right), by supplementing the static information, the model is aware of proposals that exhibit consistent static characteristics (*e.g.*, contextual environment), thereby enhancing information interaction with these proposals to *refine* them and improving the completeness of proposals. Notably, the Static-Dynamic Rectifying module is category-agnostic and can generalize to process unseen action categories.

Specifically, with the dynamic-aware action queries $\hat{\mathcal{Q}}$, we first feed them into the proposal generation head $\mathcal{F}_{gen}(\cdot)$ to obtain action proposals $\hat{t} = \sigma(\mathcal{F}_{gen}(\hat{\mathcal{Q}})) \in \mathbb{R}^{N_q \times 2}$, where σ is the sigmoid function to normalize the boundary timestamps, and $\hat{t} = \{\hat{t}_i^s, \hat{t}_i^e\}_{i=1}^{i=N_q}$. Then, the static information corresponding to the action proposals is obtained by applying temporal RoIAlign [9,28] to the static feature X extracted by CLIP, which is formulated as follows:

$$\mathcal{Z} = \text{T-RoIAlign}(\hat{t}, X) \in \mathbb{R}^{N_q \times L \times D}, \tag{3}$$

where L is the number of bins for RoIAlign. Note that the gradient backpropagation is not involved in the above process, it is only used to generate the action proposals to introduce the static information.

Subsequently, the static-dynamic action queries $\tilde{Q}$ are obtained by injecting the static features $\mathcal{Z}$ into the dynamic-aware action queries $\hat{\mathcal{Q}}$, as follows:

$$\tilde{\mathcal{Q}} = \hat{\mathcal{Q}} + \text{SA}(\text{CA}(\hat{\mathcal{Q}}, \mathcal{Z})) \in \mathbb{R}^{N_q \times D} \tag{4}$$

where the *CA* and *SA* denotes the cross-attention and self-attention, respectively. In this way, static information from different frames in $\mathcal{Z}$ is injected into the action query through attention-weighted aggregation. By injecting the static information, our action queries $\tilde{\mathcal{Q}}$ incorporate not only category-agnostic temporal dynamics from our temporal encoder but also generalizable static information from CLIP, leading to stronger cross-category detection abilities for generating complete action proposals.

3.5 Action Proposal Generation

Proposal Generation. Given the static-dynamic action queries $\tilde{\mathcal{Q}}$, we feed them into the proposal generation head $\mathcal{F}_{gen}(\cdot)$ to generate category-agnostic action proposals $\tilde{t} = \sigma(\mathcal{F}_{gen}(\tilde{\mathcal{Q}})) \in \mathbb{R}^{N_q \times 2}$, where σ is the sigmoid function to normalize the boundary timestamps, and $\tilde{t} = \{\tilde{t}_i^s, \tilde{t}_i^e\}_{i=1}^{i=N_q}$.

In addition, along with generated action proposals, we predict category-agnostic foreground probabilities $\mathcal{E} = \sigma(\mathcal{F}_{cls}(\tilde{\mathcal{Q}})) \in \mathbb{R}^{N_q}$ for action proposals, where $\mathcal{F}_{cls}$ is the binary classification head and $\mathcal{E} = \{\xi_i\}_{i=1}^{i=N_q}$.

Category-Agnostic Detection Loss. Given the action proposals $\tilde{t}$, their foreground probabilities $\mathcal{E}$ and the ground-truth action proposals $t = \{t_i^s, t_i^e\}_{i=1}^{i=N_{gt}}$.

Similar to DETR [4], we assume N_q is larger than N_{gt} and the ground-truth action proposals t is augmented to be size N_q by padding $\varnothing$. Then, the category-agnostic detection loss $\mathcal{L}_{det}$ is given as follows:

$$\mathcal{L}_{det} = \sum_{j=1}^{N_q} [\mathcal{L}_{cls}(\xi_{\hat{\pi}(j)}, \xi^*) + \mathbb{I}_{t_j \neq \varnothing} \mathcal{L}_{reg}(\tilde{t}_{\hat{\pi}(j)}, t_j)], \tag{5}$$

where $\mathcal{L}_{reg} = \mathcal{L}_1 + \mathcal{L}_{tIoU}$, and $\mathcal{L}_{cls}$ is the binary classification loss that is implemented via focal loss [27]. ξ^* is 1 if the sample is marked positive, and otherwise 0. The $\hat{\pi}$ is the permutation that assigns each ground truth to the corresponding prediction, it is obtained by Hungarian algorithm [16] as follows:

$$\hat{\pi} = \arg\min \sum_{i=1}^{N_q} Cost(\tilde{t}_i, \xi_i, t_i), \tag{6}$$

where $Cost(\tilde{t}_i, \xi_i, t_i)$ is defined as $\mathbb{I}_{\{t_i \neq \varnothing\}}[\alpha \cdot \mathcal{L}_1(\tilde{t}_i, t_i) - \beta \cdot \mathcal{L}_{tIoU}(\tilde{t}_i, t_i) - \gamma \cdot \xi_i]$, and $\mathcal{L}_{tIoU}$ is the temporal IoU loss [28].

3.6 Training Objective and Inference

Training Objective. Overall, the training objective of our GAP is given as follows:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda_{ad} \cdot \mathcal{L}_{ad}, \tag{7}$$

where $\lambda_{ad} = 3$ and the balance factor of $\mathcal{L}_{cls}$, $\mathcal{L}_1$ and $\mathcal{L}_{tIoU}$ in $\mathcal{L}_{det}$ are 3, 5 and 2, respectively.

Zero-Shot Inference. After generating the category-agnostic action proposals, following previous works [14,33], we construct the text prompt to transfer the zero-shot recognition capability of CLIP, as shown in Fig. 2 (Left).

Specifically, the category name is wrapped in a prompt template "*a video of a person doing* $< CLS >$", then the textual (*i.e.*, prompt) embeddings $\mathcal{S} \in \mathbb{R}^{N_c \times D}$ are obtained by feeding the prompt into text encoder $\mathcal{F}_t$ of CLIP, where N_c is the number of *unseen* categories.

Given the category-agnostic action proposals $\tilde{t}$ generated by the action detector, we obtain the frame features $\mathcal{Z} \in \mathbb{R}^{N_q \times L \times D}$ corresponding to action proposals by applying the temporal RoIAlign to spatial features X, as in Eq. (3). Subsequently, the action classification is conducted as follows:

$$\hat{c} = \arg\max_{c \in N_c} \psi(\cos(\mathcal{Z}, \mathcal{S})) \in \mathbb{R}^{N_q}, \tag{8}$$

where $\hat{c} = \{\tilde{c}_i\}_{i=1}^{i=N_q}$ is the set of predicted categories corresponding to the action proposals, ψ is the temporal aggregation module and $cos(\cdot, \cdot)$ denotes the cosine similarity. Subsequently, the final prediction $\mathcal{Y}_{pre} = \{\tilde{t}_i, \tilde{c}_i\}_{i=1}^{i=N_q}$ is obtained by combining the predicted action proposals $\tilde{t}$ and predicted category $\hat{c}$.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our method on two public benchmarks, *i.e.*, Thumos14 [13] and ActivityNet1.3 [2], for zero-shot temporal action localization. Following the previous methods [14,33], we adopt two split settings for zero-shot scenarios: (1) training with 75% action categories and test on the left 25% action categories; (2) training with 50% categories and test on the left 50% action categories.

Thumos14 contains 200 validation videos and 213 test videos of 20 action classes. It is a challenging benchmark with around 15.5 action instances per video and whose videos have diverse durations. We use the validation videos for training and the test videos for test, following previous works.

ActivityNet1.3 is a large dataset that covers 200 action categories, with a training set of 10,024 videos and a validation set of 4,926 videos. It contains around 1.5 action instances per video. We use the training and validation sets for training and test, respectively.

Evaluation Metric. Following previous works [14,33], we evaluate our method by mean average precision (mAP) under multiple IoU thresholds, which are standard evaluation metrics for temporal action localization. Our evaluation is conducted using the officially released evaluation code [2]. Moreover, to evaluate the quality of proposals generated by our method, we calculate Average Recall (AR) with Average Number (AN) of proposals and area under AR *v.s.* AN curve per video, which are denoted by AR@AN and AUC. Following the standard protocol [25], we use tIoU thresholds set [0.5:0.05:1.0] on Thumos14 and [0.5:0.05:0.95] on ActivityNet1.3 to calculate AR@AN and AUC.

4.2 Implementation Detatils

For a fair comparison with previous works [14,33], we *only* adopt the visual and text encoders from pre-trained CLIP [36] (ViT-B/16) to extract video and text prompt features, the dimension $D = 512$. The number of layers for the temporal encoder and decoder for Thumos14 and ActivityNet1.3 is set to 2, 5, and 2, 2 respectively. The proposal generation head, binary classification head, and temporal aggregation module are implemented by MLP, FC, and average pooling, respectively. The AdamW [29] optimizer with the batch size 16 and weight decay 1×10^{-4} is used for optimization. The equilibrium coefficients α, β and γ in Eq. (6) are specified as 5, 2 and 2. The number of bins $L = 16$ for RoIAlign. The number of action queries is set to 40 and 30, learning rate is set to 1×10^{-4} and 5×10^{-5} for Thumos14 and ActivityNet1.3. The method is implemented in PyTorch [34] and all experiments are performed on an NVIDIA GTX 1080Ti GPU. More details are available in *supplementary material.*

4.3 Comparison with State-of-the-Arts

Performance of Localization Results. In Table 1, we compare our method with the state-of-the-art ZSTAL methods on Thumos14 and ActivityNet1.3 datasets, in terms of mAP metric. From the results, it can be found that our method significantly outperforms the existing methods and achieves new state-of-the-art performance on both datasets. Our method outperforms the latest method by 3.2% and 3.4% in terms of average mAP (*i.e.*, AVG) of the 75% *v.s.* 25% split on the Thumos14 and ActivityNet1.3 datasets, respectively. In the case of the more challenging 50% *v.s.* 50% split, our method still significantly outperforms the state-of-the-art methods. This demonstrates the effectiveness of our proposed proposal-level action detector. It is worth noting that for a fair comparison with other methods, we only use CLIP (*i.e.*, RGB only) as the backbone, without the introduction of optical flow features that necessitate complex processing. This demonstrates that our GAP has excellent generalization ability to detect the location of unseen action categories by integrating generalizable static and dynamic information.

Table 1. Comparison with the state-of-the-art ZSTAL methods on Thumos14 and ActivityNet1.3 datasets. AVG represents the average mAP (%) computed under different IoU thresholds, *i.e.*, [0.3:0.1:0.7] for Thumos14 and [0.5:0.05:0.95] for ActivityNet1.3. The † denotes the extra information (*i.e.*, optical flow) is disabled for a fair comparison. All results of the compared methods are from their official report.

Split	Method	Thumos14						ActivityNet1.3			
		0.3	0.4	0.5	0.6	0.7	AVG	0.5	0.75	0.95	AVG
75% Seen 25% Unseen	DenseCLIP [37]	28.5	20.3	17.1	10.5	6.9	16.6	32.6	18.5	5.8	19.6
	CLIP [36]	33.0	25.5	18.3	11.6	5.7	18.8	35.6	20.4	2.1	20.2
	EffPrompt [14]	39.7	31.6	23.0	14.9	7.5	23.3	37.6	22.9	3.8	23.1
	STALE [33]	40.5	32.3	23.5	15.3	7.6	23.8	38.2	25.2	6.0	24.9
	ZEETAD [35]†	47.3	–	29.7	–	11.5	29.7	45.5	28.2	6.3	28.4
	Ours	**52.3**	**44.2**	**32.8**	**22.4**	**12.6**	**32.9**	**47.6**	**32.5**	**8.6**	**31.8**
50% Seen 50% Unseen	DenseCLIP [37]	21.0	16.4	11.2	6.3	3.2	11.6	25.3	13.0	3.7	12.9
	CLIP [36]	27.2	21.3	15.3	9.7	4.8	15.7	28.0	16.4	1.2	16.0
	EffPrompt [14]	37.2	29.6	21.6	14.0	7.2	21.9	32.0	19.3	2.9	19.6
	STALE [33]	38.3	30.7	21.2	13.8	7.0	22.2	32.1	20.7	5.9	20.5
	Ours	**44.2**	**36.0**	**27.1**	**15.1**	**8.0**	**26.1**	**41.6**	**26.2**	**6.1**	**26.4**

Quality of Generated Action Proposals. We conduct a comparison between our proposed GAP and existing methods in terms of the quality of generated action proposals for unseen action categories. All experiments are performed in the split 75% *v.s.* 25% on the Thumos14 dataset. Notably, the ZEETAD [35] does not release its code, so we cannot make a fair comparison with it. Following the standard protocol [25], we adopt the AR@AN and AUC as evaluation metrics, and the comparison results are summarized in table Table 3. From the results, we can find that our method significantly outperforms the previous ones in both AR and AUC metrics. This demonstrates that our GAP can generate more accurate and complete action proposals for unseen actions. This is attributed to both the proposed proposal-level detector and the integration of generalizable static and dynamic information, which significantly improves the generalizability to detect actions from unseen categories.

4.4 Analysis

We conduct extensive quantitative and qualitative analysis to demonstrate the effectiveness of our proposed GAP. All experiments are performed in the split 75% *v.s.* 25% on the Thumos14 dataset. More analyses are available in *supplementary material.*

Table 2. Ablation studies of our method on the Thumos14 dataset, adopting the 75% *v.s.* 25% split. The "Actionness" denotes the Action-aware Discrimination loss $\mathcal{L}_{ad}$ and "Rectifying" denotes the Static-Dynamic Rectifying module.

Models	mAP@IoU						AR@AN			AUC
	0.3	0.4	0.5	0.6	0.7	AVG	@10	@25	@40	
Full	**52.3**	**44.2**	**32.8**	**22.4**	**12.6**	**32.9**	**12.7**	**22.7**	**25.6**	**23.8**
w/o Rectifying	50.6	39.7	31.8	19.8	10.5	30.5	12.3	21.1	23.9	22.6
w/o Rectifying & Actionness	49.0	39.7	28.7	17.7	8.2	28.7	11.4	20.5	22.9	21.6

Table 3. Comparison with the state-of-the-art ZSTAL methods in terms of AR@AN (%) and AUC (%). "Frame" and "Proposal" denote the frame-level and the proposal-level detector, respectively.

Method	Detector Type	AR@AN			AUC
		@10	@25	@40	
EffPrompt [14]	Frame	9.3	15.7	19.6	19.3
STALE [33]	Frame	6.9	12.6	15.8	14.8
Ours	Proposal	**12.7**	**22.7**	**25.6**	**23.8**

Table 4. Comparison of different implementations of Static-Dynamic Rectifying module. All experiments are performed in the split 75% *v.s.* 25% on Thumos14.

Models	AVG	AR@AN			AUC
		@10	@25	@40	
STALE [33]	23.8	6.9	12.6	15.8	14.8
Mean	30.3	12.0	22.1	24.6	23.2
Max	31.8	12.5	21.8	24.7	23.4
Cross-Attention	**32.9**	**12.7**	**22.7**	**25.6**	**23.8**

Ablation Studies of Each Component. In Table 2, we show the quantitative analysis of the different components in our method. By comparing the first and second rows, removing the Static-Dynamic Rectifying module results in the 2.4% and 1.2% performance degradation in terms of AVG and AUC, which demonstrates that the integration of generalizable static-dynamic information does help to improve the detection abilities of the detector to generalize to unseen action categories. From the second and third rows, we find that the absence of the Action-aware Discrimination loss $\mathcal{L}_{ad}$ leads to a 1.8% and 1.0%

performance drop of AVG and AUC, respectively. This is attributed to that $\mathcal{L}_{ad}$ enhances the ability of the temporal encoder to perceive category-agnostic dynamic information. Moreover, from the third row, we find that by only adopting the category-agnostic detector, our method still outperforms the frame-level method STALE [33] 4.9% and 6.8% in terms of AVG and AUC. This is because the frame-level detector in STALE generates action proposals by grouping consecutive frames, resulting in fragmented action proposals. Our proposed proposal-level detector is able to generate action proposals directly, which guarantees the completeness of action proposals in a holistic way.

Different Implementations of Static-Dynamic Rectifying Module. In Table 4, we compare the different implementations of the Static-Dynamic Rectifying module. "Mean" and "Max" refer to the static information of different frames (*i.e.*, L) in $\mathcal{Z} \in \mathbb{R}^{N_q \times L \times D}$ aggregated through average pooling and max pooling, respectively. From the results, we find that the best performance is achieved by adopting cross-attention, which is attributed to the attention-adaptive aggregation focusing on more valuable information. Notably, regardless of different implementations, our method still outperforms the state-of-the-art method STALE [33] in all metrics. This demonstrates that combining generalizable static-dynamic information effectively improves the generalization ability of our GAP to detect unseen action categories.

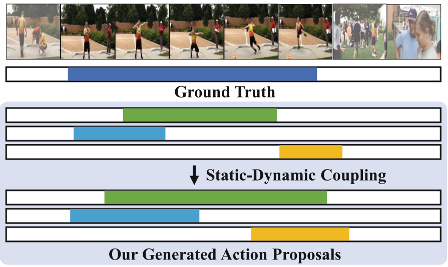


Fig. 4. Visualization of the three action proposals before and after the Static-Dynamic Rectifying module, without retraining. The same color represents the result from the same action proposal. Best viewed in color.

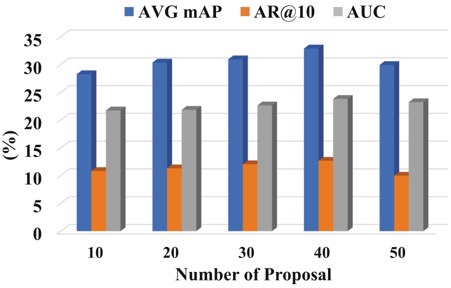


Fig. 5. Performance of different number of action queries. AVG mAP denotes the average mAP for IoU thersholds from 0.1 to 0.7 with 0.1 increment. All experiments are performed in the split 75% *v.s.* 25% on the Thumos14 dataset. Best viewed in color.

Qualitative Analysis of Static-Dynamic Rectifying. In Fig. 4, we track and visualize the changes in the specified action proposals before and after applying the Static-Dynamic Rectifying module. Note that here the *input* and *output* of the Static-Dynamic Rectifying module are compared directly, without retraining. The experiments are performed on our full method, and we choose the top-3 category-agnostic action proposals with the highest predicted scores for visualization. From the result, we find that the durations (start, end) of the three different action proposals are all refined after the Static-Dynamic Rectifying module. This further verifies that the Static-Dynamic Rectifying module improves the completeness of action proposals by exploiting the complementary nature of static-dynamic information.

Analysis of the Number of Action Queries. In Fig. 5, we compare the results under different number of action queries. Due to the query-based architecture we adopted, each action query in our action detector corresponds to an action proposal. In principle, a fewer number of action queries results in missing action instances of unseen categories, while a large number of action queries results in generating a large number of low-quality action proposals. As shown in Fig. 5, our method achieves the best performance when using a medium number of action queries (*i.e.*, 40 queries). Despite the varied performance using different numbers of action queries, our proposed GAP can outperform state-of-the-arts in all the cases as shown in the figure, which demonstrates our effectiveness in generating high-quality action proposals.

5 Conclusion

We propose a novel Generalizable Action Proposal generator named GAP, which can generate more complete action proposals for unseen action categories compared with previous works. Our GAP is designed with a query-based architecture, enabling it to generate action proposals in a holistic way. The GAP eliminates the need for hand-crafted post-processing, supporting seamless integration with CLIP to solve ZSTAL. Furthermore, we propose a novel Static-Dynamic Rectifying module, which integrates generalizable static and dynamic information to improve the completeness of action proposals for unseen categories. Extensive experiments on two datasets demonstrate the effectiveness of our method, and our approach significantly outperforms previous methods, achieving a new state-of-the-art performance.

Acknowledgements. This work was supported partially by NSFC (No. 62206315), Guangdong NSF Project (No. 2023B1515040025, No. 2024A1515010101), Guangzhou Basic and Applied Basic Research Scheme (No. 2024A04J4067).

References

1. Buch, S., Eyzaguirre, C., Gaidon, A., Wu, J., Fei-Fei, L., Niebles, J.C.: Revisiting the "video" in video-language understanding. In: CVPR (2022)
2. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: a large-scale video benchmark for human activity understanding. In: CVPR (2015)
3. Cao, M., Yang, T., Weng, J., Zhang, C., Wang, J., Zou, Y.: LocVTP: video-text pre-training for temporal localization. In: ECCV (2022)

4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
5. Cheng, F., Wang, X., Lei, J., Crandall, D., Bansal, M., Bertasius, G.: VindLU: a recipe for effective video-and-language pretraining. In: CVPR (2023)
6. Deng, C., Chen, Q., Qin, P., Chen, D., Wu, Q.: Prompt switch: efficient CLIP adaptation for text-video retrieval. In: ICCV (2023)
7. Du, J.R., et al.: Weakly-supervised temporal action localization by progressive complementary learning. arXiv (2022)
8. Feng, J.C., Hong, F.T., Zheng, W.S.: MIST: multiple instance self-training framework for video anomaly detection. In: CVPR (2021)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
10. Hong, F.T., Feng, J.C., Xu, D., Shan, Y., Zheng, W.S.: Cross-modal consensus network for weakly supervised temporal action localization. In: ACM MM (2021)
11. Hong, F.T., Huang, X., Li, W.H., Zheng, W.S.: MINI-Net: multiple instance ranking network for video highlight detection. In: ECCV (2020)
12. Huang, J., Li, Y., Feng, J., Wu, X., Sun, X., Ji, R.: Clover: towards a unified video-language alignment and fusion model. In: CVPR (2023)
13. Jiang, Y.G., et al.: THUMOS challenge: action recognition with a large number of classes (2014). http://crcv.ucf.edu/THUMOS14/
14. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV (2022)
15. Ju, C., et al.: Multi-modal prompting for low-shot temporal action localization. arXiv (2023)
16. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. (1955)
17. Li, D., Li, J., Li, H., Niebles, J.C., Hoi, S.C.: Align and prompt: video-and-language pre-training with entity prompts. In: CVPR (2022)
18. Li, Y.M., Huang, W.J., Wang, A.L., Zeng, L.A., Meng, J.K., Zheng, W.S.: EgoExo-Fitness: towards egocentric and exocentric full-body action understanding. In: ECCV (2024)
19. Li, Y.M., Zeng, L.A., Meng, J.K., Zheng, W.S.: Continual action assessment via task-consistent score-discriminative feature distribution modeling. TCSVT (2024)
20. Lin, C., et al.: Learning salient boundary feature for anchor-free temporal action localization. In: CVPR (2021)
21. Lin, K.Q., et al.: UniVTG: towards unified video-language temporal grounding. In: ICCV (2023)
22. Lin, K.Y., et al.: Rethinking CLIP-based video learners in cross-domain open-vocabulary action recognition. arXiv (2024)
23. Lin, K.Y., Du, J.R., Gao, Y., Zhou, J., Zheng, W.S.: Diversifying spatial-temporal perception for video domain generalization. In: NeurIPS (2024)
24. Lin, K.Y., Zhou, J., Zheng, W.S.: Human-centric transformer for domain adaptive action recognition. TPAMI (2024)
25. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: BMN: boundary-matching network for temporal action proposal generation. In: ICCV (2019)
26. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. In: ECCV (2018)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
28. Liu, X., et al.: End-to-end temporal action detection with transformer. TIP (2022)
29. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2017)

30. Luo, D., Huang, J., Gong, S., Jin, H., Liu, Y.: Towards generalisable video moment retrieval: visual-dynamic injection to image-text pre-training. In: CVPR (2023)
31. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: learning a text-video embedding by watching hundred million narrated video clips. In: ICCV (2019)
32. Moon, W., Hyun, S., Park, S., Park, D., Heo, J.P.: Query-dependent video representation for moment retrieval and highlight detection. In: CVPR (2023)
33. Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Zero-shot temporal action detection via vision-language prompting. In: ECCV (2022)
34. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: NeurIPS (2019)
35. Phan, T., Vo, K., Le, D., Doretto, G., Adjeroh, D., Le, N.: ZEETAD: adapting pretrained vision-language model for zero-shot end-to-end temporal action detection. In: WACV (2024)
36. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
37. Rao, Y., et al.: DenseCLIP: language-guided dense prediction with context-aware prompting. In: CVPR (2022)
38. Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., Tao, D.: TriDet: temporal action detection with relative boundary modeling. In: CVPR (2023)
39. Shi, D., et al.: ReAct: temporal action detection with relational queries. In: ECCV (2022)
40. Sun, S., Gong, X.: Hierarchical semantic contrast for scene-aware video anomaly detection. In: CVPR (2023)
41. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: ICCV (2021)
42. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
43. Wang, A.L., Lin, K.Y., Du, J.R., Meng, J., Zheng, W.S.: Event-guided procedure planning from instructional videos with text supervision. In: ICCV (2023)
44. Wu, W., Luo, H., Fang, B., Wang, J., Ouyang, W.: Cap4Video: what can auxiliary captions do for text-video retrieval? In: CVPR (2023)
45. Xu, H., et al.: VideoCLIP: contrastive pre-training for zero-shot video-text understanding. arXiv (2021)
46. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-TAD: sub-graph localization for temporal action detection. In: CVPR (2020)
47. Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: CVPR (2016)
48. Zhang, C., et al.: Exploiting completeness and uncertainty of pseudo labels for weakly supervised video anomaly detection. In: CVPR (2023)
49. Zhang, C.L., Wu, J., Li, Y.: ActionFormer: localizing moments of actions with transformers. In: ECCV. Springer (2022)
50. Zhou, J., Liang, J., Lin, K.Y., Yang, J., Zheng, W.S.: ActionHub: a large-scale action video description dataset for zero-shot action recognition. arXiv (2024)
51. Zhou, J., Lin, K.Y., Li, H., Zheng, W.S.: Graph-based high-order relation modeling for long-term action recognition. In: CVPR (2021)
52. Zhou, J., Lin, K.Y., Qiu, Y.K., Zheng, W.S.: TwinFormer: fine-to-coarse temporal modeling for long-term action recognition. TMM (2023)

GRAtt-VIS: Gated Residual Attention for Video Instance Segmentation

Tanveer Hannan[1,2(✉)], Rajat Koner[1,2], Maximilian Bernhard[1,2], Suprosanna Shit[3], Bjoern Menze[4], Volker Tresp[1,2], Matthias Schubert[1,2], and Thomas Seidl[1,2]

[1] LMU Munich, Munich, Germany
[2] MCML, Munich, Germany
{hannan,koner}@dbs.ifi.lmu.de
[3] Technical University of Munich, Munich, Germany
[4] University of Zurich, Zürich, Switzerland

Abstract. Video Instance Segmentation (VIS) has seen a growing reliance on query propagation-based methods to model complex and lengthy videos. While these methods dominate the performance, they do not explicitly model discrete events, e.g., occlusion, disappearance, and reappearance. Such events often results in degraded object features over time. We believe learning these events end-to-end with the propagation network would prevent the degradation. To this end, we propose a novel propagation method that models these discrete events with a gating mechanism. First, the gate identifies degraded object features caused by these events. Second, we apply a residual configuration to rectify the feature degradation, alleviating the need for a conventional memory bank. Third, we restrict interaction between relevant and degraded objects with a novel gated self-attention. The gated residual configuration and self-attention forms **GRAtt** block, which can easily be integrated into the existing propagation frameworks. GRAtt-VIS performs on par with the state-of-the-art methods on YTVIS-19,-21,-22 and challenging OVIS datasets by significantly improving performance over previous methods. The code is available in the supplementary.

Keywords: Video Instance Segmentation · Multi Object Tracking

1 Introduction

Video Instance Segmentation (VIS) [26] is a complex task that requires detecting, segmenting, and tracking all instances or objects within a video sequence. Existing methodologies for VIS can be broadly categorized into offline and online

T. Hannan and R. Koner—Joint first-authorship and equal contributions.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_18.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 268–282, 2025.
https://doi.org/10.1007/978-3-031-78444-6_18

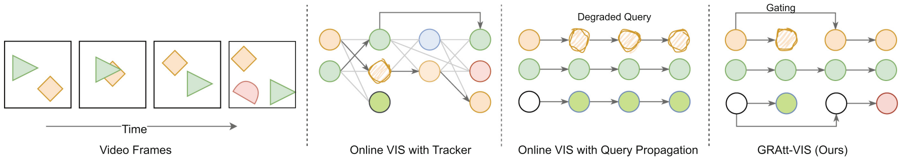


Fig. 1. While tracker-based methods suffer from computational complexity, vanilla propagation lacks a decision system. GRAtt-VIS bridges both paradigms with a single network capable of replacing the heuristic association of the tracker with a gated propagation method.

methods. Offline methods [3,7,18,22,29] process the entire video at once. In contrast, online methods [4,6,8,11,12,21,23,28] process video sequences frame by frame. The recent emergence of datasets [17,25] containing lengthy and occluded videos has presented more challenging, real-world scenarios for VIS. Notably, online-VIS models have shown remarkable robustness in processing these lengthy and challenging videos and achieved higher precision than offline approaches.

As illustrated in Fig. 1, these models primarily rely on frame-level detection and inter-frame association facilitated either through a tracker [4,8,23,27] or a propagation method [6,11,21,30]. Tracker-based methods employ heuristic algorithms to tackle disruptive events, e.g., occlusion and reappearance. First, they match objects in the present frame with the detected objects of the past ones. This similarity matching removes duplicate detection and improves robustness against partial occlusion. Second, during object disappearance, the tracker does not update instance representation. Instead, it matches the reappearing object with the representation of the last visible one. Third, tracker-based methods selectively keep object representation between frames, allowing more free slots for newly appearing objects. However, their reliance on heuristic algorithms restricts end-to-end learning, decreases inference speed, and lacks scalability across datasets.

On the other hand, the query propagation-based methods became alternative choices because of their simplicity and less heuristic dependency. However, they **lack an explicit decision system** to model the discrete events in the video. For example, in the case of occlusion, it is necessary to identify and prevent the accumulation of degraded features. Instead, the current query propagation-based methods continually accumulate noisy object representations even during occlusion or disappearance and do not possess any mechanism to recover from the degraded features. Few methods like [6,11] incorporate a memory bank consisting of past representations to mitigate the impact of erroneous propagation. However, designing an optimal memory queue is a double-edged sword because a small memory size may not facilitate instance recovery, while a large memory may introduce noisy representations. Moreover, integrating a memory queue through cross-attention is resource-intensive and poses optimization challenges due to the 'irrelevant' and redundant features in the memory bank.

Although query propagation can implicitly learn these shocks in a data-driven manner, we postulate injecting a **discrete decision system as an inductive bias** into the network is beneficial. Ideally, such a decision system should trigger automatically and immediately in cases of *shock*, e.g., blur, abrupt camera movement, occlusion, disappearance, and new object appearance. In the presence of shock, instance queries lose their context and may become *irrelevant queries*, whereas previous frame queries remain more useful for future detection. Our *first objective* is to detect such degraded query representations whenever a shock occurs. Finding hand-crafted criteria for this binary detection is hard to formalize exhaustively. Instead, we let the model decide which query is relevant and which one is degraded without any explicit supervision. Our *second objective* is to prevent the propagation of degraded queries and allow relevant and unallocated ones from past frames. The unallocated queries could facilitate new object detection. Our *third objective* is to preserve the relevant queries from interacting with the degraded ones.

Our Contribution: To this end, we present a Gated Residual Attention for Video Instance Segmentation, termed GRAtt-VIS as shown in Fig. 2. Our approach aims to explicitly identify and model discrete events in a video to enhance temporal consistency in query propagation while making it robust against abrupt noise or shock. **Firstly**, we introduce a gating mechanism that learns to predict whether the current query is relevant or degraded. **Secondly**, we use the gating signal in a residual configuration that controls query propagation conditioned on the learned relevance. This prevents the degraded queries from accumulating noise that could affect current and future predictions. **Thirdly**, to preserve the relevant queries from interacting with the degraded ones, we introduce a novel gated self-attention. Additionally, we simplify the VIS pipeline by eliminating complex and expensive memory mechanisms.

Together, these components improve on key challenges of VIS: occlusion, new object detection, and robustness against abrupt shock in the video. Importantly, our method can be integrated ad-hoc into existing propagation-based networks, e.g., InstanceFormer [11] or GenVIS [6], emphasizing its generalizability. GRAtt-VIS performs on par with the state-of-the-art methods across multiple benchmark VIS datasets, such as, YTVIS-19, -21, -22 [24–26] and OVIS [17]. Compared to the prior baseline [6], GRAtt-VIS improves performance in Average Precision (AP) by 1.8% in YTVIS-21, 3.3% on YTVIS-22 long videos, and 0.4% on OVIS.

2 Related Literature

Offline-VIS processes the whole video simultaneously, making future frames available during inference. Earlier offline-VIS incorporated instance mask propagation [1,13] for temporal connection. Recently, the instance query of the Detection Transformer [2] has been exploited vastly in this paradigm. VisTR [20] pioneered this line of research with the very first end-to-end trainable query-based method. IFC [9], SeqFormer [22], continued this effort by reducing overall complexity and improving performance. They focused on building efficient temporal attention mechanisms to achieve this feat. Mask2Former-VIS [3] further demonstrated that powerful frame-level query-based detectors could compete with contemporary methods with little overhead. TeViT [29] and VITA [7] went further by limiting the temporal attention with a more effective shifted window mechanism. Finally, EfficientVIS [21] developed a streamlined heuristic for linking clips, thereby facilitating the processing of longer videos.

Tracker-based Online-VIS requires heuristic post-processing on top of network detection. MaskTrack R-CNN [26] associates instances with a track head during inference. Subsequent tracker-based works, CrossVIS [27], and VISOLO [4] improved on top of it by utilizing video-level properties in their training pipeline. Though trained on frame-level, MinVIS [8] substantially improves on previous methods by leveraging a powerful object detector Mask2Former [3] and tracking instances with bipartite matching. The current state-of-the-art tracker-based method IDOL [23], is built on Deformable-DETR [31]. It adopts contrastive learning on the instance queries between frames during training and deploys a heuristic instance matching during inference.

Propagation-based Online-VIS eliminates the need for an external tracker or data association by leveraging the instance queries readily available in Detection Transformer [2] based architectures. TrackFormer [16] initially demonstrated this powerful technique on Multi-Object Tracking and Segmentation [19] challenge. Later on, this method was adopted by InstanceFormer [11], and ROVIS [30] for the VIS datasets. InstanceFormer propagated instance queries and reference points of Deformable-DETR [31] from frame to frame. On the other hand, ROVIS utilizes the more powerful frame level Mask2Former [3] features to establish the inter-frame link. Before our work, GenVIS [6] held state-of-the-art performance by crafting a decisive video-level training strategy utilizing the Mask2Former architecture.

3 Methodology

In this section, we present GRAtt-VIS, comprising a novel Gated Residual Attention (GRAtt) in the Decoder. We begin by providing a brief background on the query propagation-based VIS methods. Afterward, we present our proposed gating mechanism, its use in residual configuration, and gated self-attention.

3.1 Background: Query Propagation Based VIS

Query Propagation methods [6,11], use a transformer to process each frame and pass their instance representation to the next one. Let video $\mathbf{X} \in \mathbb{R}^{N_f \times H \times W \times 3}$ consist of N_f frames of height H and width W. At the time t, a query propagation-based model extracts feature f_t from a frame $x_t \in \mathbf{X}$ through its feature extractor $f_t = \Phi(x_t)$. Afterward, it contextualizes f_t through a transformer

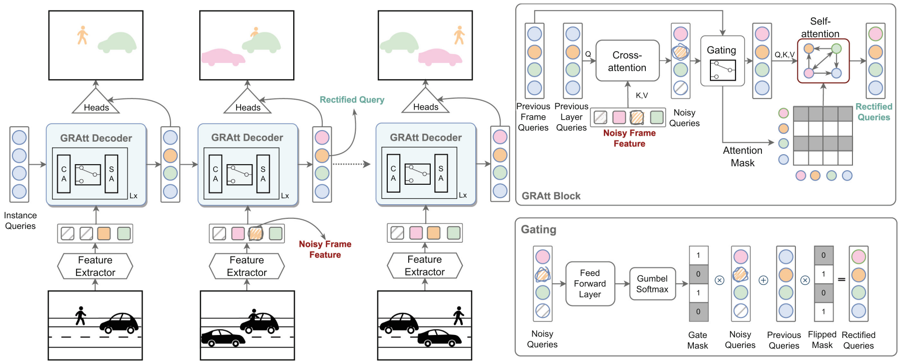


Fig. 2. The architecture of GRAtt-VIS. Our GRAtt-decoder is a generic architecture to make the temporal query propagation robust and stable against abrupt noise and shock. This is achieved by a gated residual connection following a masked self-attention. Together they form a Gated Residual Attention (GRAtt) block, which learns to rectify the effect of noisy features and implicitly preserves 'relevant' instance representations along the temporal dimension. The GRAtt block is a simple replacement for computation-heavy memory and provides superior performance on complex videos across multiple propagation-based VIS frameworks.

encoder to be used by a decoder. The decoder uses N instance queries that learn the instance representation.

In the decoder D, the instance queries $\{q_t^i\}_{i=1}^N$ attend to current frame features f_t through multiple layers of cross-attention followed by a self-attention-based contextualization. After processing the current frame, the instance queries are propagated to the subsequent frame as in Eq. 1.

$$q_{t+1}^i = D(q_t^i, f_t) \tag{1}$$

The propagation mechanism assigns instances with a particular indexed query throughout the video. Once a query has been associated with an instance, it becomes persistently linked to that query, thereby preventing reassignment to any new objects that may appear later in the video. The unallocated queries capture a new object's appearance. During training, Hungarian matching is applied only upon the appearance of a new object. To mitigate noise and re-identify objects in long videos, these models [6,11] often employ additional memory attention. This feature is implemented by an extra cross-attention module and a memory bank consisting of past instance queries. Recent works [6,11,30] follow the query propagation and memory queue as the main principle.

3.2 GRAtt Decoder

At the core of our contribution lies a gating mechanism within the decoder layers. We aim to assess the relevance of queries to correctly capture instance representation and discard them in case of degradation. This is a discrete decision-making process ('yes' or 'no') necessitating a gating mechanism that provides a binary decision within the decoder. We leverage the Gumbel-Softmax trick [10,15], a technique capable of transforming a continuous distribution into a categorical one. This trick allows us to attain the binary gating output while ensuring end-to-end differentiability. Our gating mechanism learns the distribution of occurrences of abrupt perturbations based on a dataset. We apply this gating mechanism at every decoder layer to allow multiple checks against the degraded features. Note that we do not provide any explicit supervision to the gating output. Consequently, one can interpret the gate output as an auto-rectifying mechanism targeted toward the most useful instance queries throughout the temporal dynamics.

Discrete Gating: At frame t, let $q_t^{i,l} \in \mathbb{R}^C$ be the i^{th} input object query of l^{th} decoder layer, where where C is the query dimension, $i \in \{1,..,N\}$ and $l \in \{1,..,L\}$. We place the gating between each cross- and self-attention layer. We want to assess the relevancy of the query feature with respect to the current frame. At decoder layer l, after the cross-attention, for $q_t^{i,l}$ we obtained its corresponding gate signal $g_t^{i,l}$ as

$$g_t^{i,l} = f_{\mathrm{g}}\left(q_t^{i,l}\right), \tag{2}$$

Here f_{g} is a linear projection layer of output dimension 1. To obtain a categorical variable $G_t^{i,l}$ with probabilities ${\pi_1}_t^{i,l} = \sigma(g_t^{i,l})$ and ${\pi_0}_t^{i,l} = 1 - \sigma(g_t^{i,l})$, where σ is the sigmoid operation. We can reparameterize the sampling process of $G_t^{i,l}$ using the Gumbel-Max trick as follows:

$$G_t^{i,l} = \arg\max_k \left\{\log\left({\pi_k}_t^{i,l}\right) + g_k : k = 0, 1\right\} \tag{3}$$

Here, $\{g_k\}_{k=0,1}$ are i.i.d. random variables sampled from $Gumbel(0,1)$. Due to the discontinuous nature of the argmax operation, we approximate $G_t^{i,l}$ with a differentiable, soft version $\hat{G}_t^{i,l}$, obtained from the Gumbel-Softmax relaxation:

$$\hat{G}_t^{i,l} = \frac{\exp\left(\left(\log\left(\pi_1 t^{i,l}\right) + g_1\right)/\tau\right)}{\Sigma_{k\in 0,1}\exp\left(\left(\log\left({\pi_k}_t^{i,l}\right) + g_k\right)/\tau\right)} \tag{4}$$

Following the recommendations in the literature [15], we set the softmax temperature τ to 0.67. Finally, to achieve differentiability with the discrete samples $G_t^{i,l}$, we apply the Straight-through trick [10] and use the gradients of $\hat{G}_t^{i,l}$ as an approximation for the gradients of $G_t^{i,l}$ in the backward pass.

Residual Configuration: By convention, instance queries flagged as '0' in the gating mechanism are considered 'irrelevant' and do not propagate through the current frame. If propagated, the 'irrelevant' features could accumulate erroneous contextualization of instances. Instead, a residual connection from the previous frame supplies a relevant representation of instance queries. The gated residual connection is placed between the cross-attention and self-attention of l^{th} decoder layer D_l. The output of l^{th} layer ($q_t^{i,l+1}$) can be described as

$$q_t^{i,l+1} = \begin{cases} D_l(q_t^{i,l}, f_t) & \text{if } G_t^{i,l} = 1 \\ q_{t-1}^{i,L+1} & \text{if } G_t^{i,l} = 0 \end{cases} \tag{5}$$

Equation 5 allows propagation of all 'relevant' queries to the subsequent layers, whereas degraded queries are rectified with their preceding temporal counterpart. Such residual connection effectively manages the flow of relevant queries in a video in cases of occlusion and object disappearance and reappearance. For example, if an instance is occluded at frame t, our proposed gating mechanism could preserve the information from frame $t-1$ if the gate activation stays '0'. If the object reappears at $t+10$, the preserved query could be used to re-identify the same object. This also alleviates the need for a memory bank of past representation to recover from shocks in the current frame.

Although the residual connection is placed between layers, the correction term, $q_{t-1}^{i,L+1}$ is retrieved from the past frame instead of the past layer. This configuration establishes effective frame-to-frame continuation and provides additional expressiveness during intra-frame processing. Consequently, degraded features are rectified through the residual connection with the 'relevant' ones in case of occlusion or distortion. Furthermore, unallocated instance queries might be available during processing and can easily bind to new objects. The ablation on different design choices of the place of the residual connection among inter-frame, inter-layer, and inter-attention is visualized in Fig. 7 and justified in ablation Table 6.

Gated Self-attention: In this subsection, we use italic letters to represent *query*, *key*, and *value* of the gated self-attention layers. Upon computation of $G_t^{i,l}$ for each query, we propose a novel *gated self-attention* with a query selector for the attention layers. This selector allows object queries with a corresponding gate value of '1' to engage in global attention with the rest of the queries. In contrast, object queries yielding a '0' gate output are removed from the *query* set, effectively removing them from self-attention considerations.

A gate value of '0' indicates irrelevant frame features for a particular instance query. Obtaining additional information from these irrelevant queries inside the self-attention is unlikely. Further, there is the possibility of noisy frame feature injection through self-attention. Therefore, by removing such queries, we adopt a greedy strategy to prefer undistorted instance representations over the recently degraded ones. Conversely, global attention to 'relevant' queries with gate value '1' enables them to contextualize useful information from other queries.

Let's denote $\boldsymbol{I}_t^l = [q_t^{1,l}, q_t^{2,l}, \cdots q_t^{N,l}] \in \mathbb{R}^{N \times C}$ to be the complete set of all instance queries at the l^{th} decoder layer of t^{th} frame. We define the relevant instance queries as $\tilde{\boldsymbol{I}}_t^l = \{q_t^{i,l} \in \boldsymbol{I}_t^l \mid G_t^{i,l} = 1\} \in \mathbb{R}^{M \times C}$. The degraded queries are defined as $\boldsymbol{I}_t^l = \boldsymbol{I}_t^l \backslash \tilde{\boldsymbol{I}}_t^l \in \mathbb{R}^{(N-M) \times C}$

The *queries*, denoted as $\boldsymbol{Q} = f_Q(\tilde{\boldsymbol{I}}_t^l) \in \mathbb{R}^{M \times C}$ are computed from the relevant features. In contrast, the *keys* and *values*, denoted as $\boldsymbol{K} = f_K(\boldsymbol{I}_t^l) \in \mathbb{R}^{N \times C}$ and

$\boldsymbol{V} = f_V(\boldsymbol{I}_t^l) \in \mathbb{R}^{N \times C}$ are computed from all the instance features. Here, $f_Q(\cdot)$, $f_K(\cdot)$, and $f_V(\cdot)$ denote the projection layers for the *query*, *key*, and *value* respectively.

The *gated self-attention* is computed as

$$\begin{aligned}\tilde{\boldsymbol{I}}_t^{l+1} &= \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V} + \tilde{\boldsymbol{I}}_t^l \\ \boldsymbol{I}_t^{l+1} &= [\tilde{\boldsymbol{I}}_t^{l+1}, \hat{\boldsymbol{I}}_t^l]\end{aligned} \tag{6}$$

The final concatenation of relevant, $\tilde{\boldsymbol{I}}_t^{l+1}$ and degraded queries, $\hat{\boldsymbol{I}}_t^l$ in Eq. 6 preserves the original order of the query index. We justify our design choice in comparison with other attention configurations (c.f. Fig. 8) in Table 7 in the

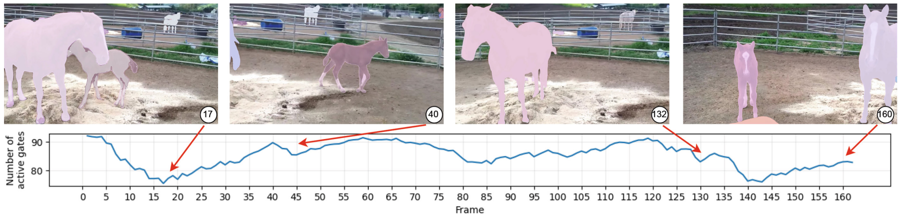


Fig. 3. Number of active gates across frames. When instances queries undergo shock, in this example, disappearance and reappearance, the number of active gates drops. In the last displayed frame, many gates have turned active again, as the objects are clearly visible.

Table 1. Quantitative performance of GRAtt-VIS compared to previous methods on YTVIS-19/21 datasets. GRAtt-VIS achieves significant improvement on both datasets with ResNet-50 backbone.

Method		YouTube-VIS 2019					YouTube-VIS 2021				
		AP	AP$_{50}$	AP$_{75}$	AR$_1$	AR$_{10}$	AP	AP$_{50}$	AP$_{75}$	AR$_1$	AR$_{10}$
Offline	EfficientVIS [21]	37.9	59.7	43.0	40.3	46.6	34.0	57.5	37.3	33.8	42.5
	IFC [9]	41.2	65.1	44.6	42.3	49.6	35.2	55.9	37.7	32.6	42.9
	Mask2Former-VIS [3]	46.4	68.0	50.0	-	-	40.6	60.9	41.8	-	-
	TeViT [29]	46.6	71.3	51.6	44.9	54.3	37.9	61.2	42.1	35.1	44.6
	SeqFormer [22]	47.4	69.8	51.8	45.5	54.8	40.5	62.4	43.7	36.1	48.1
	VITA [7]	49.8	72.6	54.5	**49.4**	**61.0**	45.7	67.4	49.5	**40.9**	53.6
Online	CrossVIS [27]	36.3	56.8	38.9	35.6	40.7	34.2	54.4	37.9	30.4	38.2
	VISOLO [4]	38.6	56.3	43.7	35.7	42.5	36.9	54.7	40.2	30.6	40.9
	ROVIS [30]	45.5	63.9	50.2	41.8	49.5	-	-	-	-	-
	InstanceFormer [11]	45.6	68.6	49.6	42.1	53.5	40.8	62.4	43.7	36.1	48.1
	MinVIS [8]	47.4	69.0	52.1	45.7	55.7	44.2	66.0	48.1	39.2	51.7
	IDOL [23]	49.5	**74.0**	52.9	47.7	58.7	43.9	68.0	49.6	38.0	50.9
	GenVIS [6]	50.0	71.5	54.6	**49.5**	**59.7**	47.1	67.5	51.5	41.6	54.7
	GRAtt-VIS (Ours)	**50.4**	70.7	**55.2**	48.4	58.7	**48.9**	**69.2**	**53.1**	**41.8**	**56.0**

supplementary document. Together, the gated residual connection and self-attention maintain the integrity and temporal consistency of object representation, enhancing our model's robustness and performance. Finally, similar to previous methods [6, 11], the object queries go through a classification and mask head for the final segmentation.

4 Experiments

We compare GRAtt-VIS with the state-of-the-art models on YTVIS and OVIS. We achieve competitive performance on all benchmarks, outperforming the previous methods.

Table 2. Performance comparison of GRAtt-VIS with recently developed VIS frameworks on the most challenging Occluded (OVIS) and Long (YTVIS-22) Video Instance Segmentation data-sets. The evaluation of SeqFormer is taken from IDOL.

Method		OVIS					YouTube-VIS 2022				
		AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Offline	SeqFormer [22]	15.1	31.9	13.8	10.4	27.1	–	–	–	–	–
	TeViT [29]	17.4	34.9	15.0	11.2	21.8	–	–	–	–	–
	VITA [7]	19.6	41.2	17.4	11.7	26.0	**32.6**	**53.9**	**39.3**	**30.3**	**42.6**
Online	CrossVIS [27]	14.9	32.7	12.1	10.3	19.8	–	–	–	–	–
	VISOLO [4]	15.3	31.0	13.8	11.1	21.7	–	–	–	–	–
	InstanceFormer [11]	20.0	40.7	18.1	12.0	27.1	32.0	55	34.5	29.5	38.3
	MinVIS [8]	25.0	45.5	24.0	13.9	29.7	33.1	54.8	33.7	29.5	36.6
	IDOL [23]	30.2	51.3	30.0	15.0	37.5	–	–	–	–	–
	ROVIS [30]	30.2	53.9	30.1	13.6	36.3	–	–	–	–	–
	GenVIS [6]	35.8	**60.8**	36.2	16.3	39.6	37.5	**61.6**	41.5	32.6	42.2
	GRAtt-VIS (Ours)	**36.2**	**60.8**	**36.8**	**16.8**	**40.0**	**40.8**	60.1	**45.9**	**35.7**	**46.9**

Datasets: We experimented on four benchmark datasets, YouTubeVIS(YTVIS)-19/21/22 [24, 25] and OVIS [17]. YTVIS is an evolving dataset with three iterations from 2019, 2021, and 2022. It focuses on segmenting and tracking video objects with 40 predefined categories. The dataset's complexity has increased with the introduction of more intricate, longer videos containing complex object trajectories. Despite the YTVIS 2021 and 2022 versions sharing an identical training set, additional 71 long videos have been introduced to the validation set of the 2022 version. In comparison, OVIS is a more recent dataset comprising 25 specified categories, presenting a more challenging setup with significantly higher occlusion.

Implementation Details: GRAtt-VIS uses a pre-trained frozen Mask2-Former [3] backbone to extract frame-wise features from a given video. We train GRAtt-VIS with AdamW [14] optimizer with a learning rate of $5 * 10^{-5}$ for 140K iterations. We train our model with five frames, which we sample randomly. The frames are sampled five to fifteen time-steps apart from each other. We use random flipping and cropping as data augmentation techniques. Following standard practice [6,11,30], we also use augmented COCO to supplement the primary dataset like OVIS or YTVIS to increase the number of samples for training. We train our model on four Nvidia-RTX-A6000 GPUs with a batch size of eight.

4.1 Main Results

YTVIS 19 & 21: Table 1, compares the performance of GRAtt-VIS with previous methods on both YTVIS-19 and 21 datasets. We observe comparable performance for YTVIS-19 and attribute it to label inaccuracy which was improved in the 2021 version. Moreover, YTVIS-21 added more challenging videos on top of YTVIS-19, which contains mostly short videos with gradual changes. GRAtt-VIS outperforms the prior query propagation based GenVIS by 1.8% AP.

YTVIS 22: Table 2, illustrates the performance of GRAtt-VIS on the YTVIS-22 dataset. GRAtt-VIS improves the overall AP by 3.3% on top of GenVIS, thereby setting a new benchmark. Note that YTVIS-22 is the most challenging dataset in YTVIS, containing longer and more complex videos. The enhanced performance on longer videos also validates the efficacy of the proposed residual propagation and masked self-attention strategy in preserving the relevant object features throughout long video sequences.

OVIS: Lastly, we evaluate our method on the highly challenging OVIS dataset in Table 2. As in YTVIS-22, we observe a similar trend in GRAtt-VIS's superior performance. We achieve a 0.4% improvement over GenVIS, which has a dedicated memory bank. Compared to GenVIS without this memory bank, we observe 1.3% gain in Tab. 3, suggesting our proposed propagation mechanism is a better and more lightweight alternative to the memory-based methods.

4.2 Ablation Studies

We present ablation studies to show the proposed modules' effectiveness and generalizability. We empirically illustrate the underlying mechanics of the gate. Here, we use the ResNet-50 [5] backbone and the OVIS [17] dataset.

Table 3. Cumulative Ablation for Baseline Memory and the proposed Discrete Gate, Residual Configuration, and Gated Attention. Detailed ablation for each component is included in the supplementary (Sec. A). $*$ denotes the memory-free baseline.

Model	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Memory Baseline	35.4	60.2	36.0	16.3	39.8
GRAtt-VIS	**36.2**	**60.8**	**36.8**	**16.8**	**40.0**
w/o Gated Attn.	35.2	58.6	35.5	16.7	39.0
w/o Residual Config.*	34.9	57.6	36.3	16.3	36.3
w/o Discrete Gate	34.2	57.7	33.5	16.2	39.5

Effects of Proposed Modules: Table 3 provides a comprehensive ablation analysis on the OVIS dataset, examining various components of GRAtt-VIS. Our baseline models include both the memory-free and memory-inclusive variants of GenVIS [6], which we reproduce for our ablation. Notably, GRAtt-VIS demonstrates a remarkable performance improvement, surpassing the memory-free baseline by 1.3% in AP and even outperforming the memory-inclusive baseline by 0.8% in AP. The sequential propagation of past instances facilitated by the gated residual connection maintains instance representations amidst shocks or noise in the current frame. Consequently, this eliminates the need for a memory module, reducing complexity and mitigating optimization challenges.

Notably, the performance improvement was complemented by reducing Decoder GFLOPs from 3.7 to 2.4. This reduction in GFLOPs was achieved by eliminating the memory module and subsetting the queries in the Decoder self-attention. Instances with a gated output of '0' do not require feature updates; therefore, their past features are reused. This reduction of computational complexity and elimination of memory places GRAtt-VIS in a better operating point in the efficiency vs. performance landscape.

Moreover, we also ablate between our choice of discrete gating mechanism against soft-gating. Table 3 verifies that discrete Gumble-Softmax gating improves the AP by 2% in comparison to soft-gating. Therefore, in the presence of shock, a definitive 'yes' or 'no' choice between noisy and contextual features is helpful for VIS modeling. For instance, if a shock perturbs an object in the current frame, the corresponding query will also be distorted. With soft gating, the noise would still be partially present if we chose a weighted sum of a noisy query and a relevant one. Moreover, the sparsification of the decoder self-attention, and simplification of the VIS pipeline by eliminating a memory module help the GRAtt decoder converge 33% faster than the baselines. We further include detailed ablation for each proposed module in Sec. A of the supplementary.

Table 4. Impact of GRAtt decoder integrated with InstanceFormer. It improves on both the memory-free and memory-based variants. $*$ denotes the memory-free baseline

Decoder Type	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Memory Baseline	20.0	40.7	18.1	12.0	27.1
GRAtt-VIS (Ours)	**22.3**	**43.0**	**19.5**	**12.1**	**29.8**
w/o Gating*	17.1	35.5	15.4	9.8	24.7

Generizability of GRAtt Decoder: The SOTA propagation methods [3,6,30] utilize Masked Attention. To show the universality of GRAtt decoder, we test it on a different setup, e.g., Deformable Attention of InstanceFormer in Tab. 4. We replaced its decoder layer with our GRAtt module. The GRAtt module also outperforms its memory-free and memory baseline by 5.2% and 2.3% AP.

Temporally Consistency: We visualize query features across video frames through t-SNE plots, comparing the baseline model GenVIS with our proposed

GRAtt-VIS in Fig. 4. Our gating mechanism effectively prevents query degradation, leading to greater temporal consistency in instance features. This consistency fosters discriminative features, resulting in distinct clusters of instances in the feature space. Consequently, this improvement enhances the overall performance of GRAtt-VIS.

Gate Statistics: The gate activation statistics provide insight into how gating behaves in the presence of many instances or occlusion. Figure 5 (left) shows the average gate activations decrease with more objects. This phenomenon aligns with the higher occurrence of occlusions and object disappearances in scenarios with more objects. The decrease in gate activations signifies that gat-

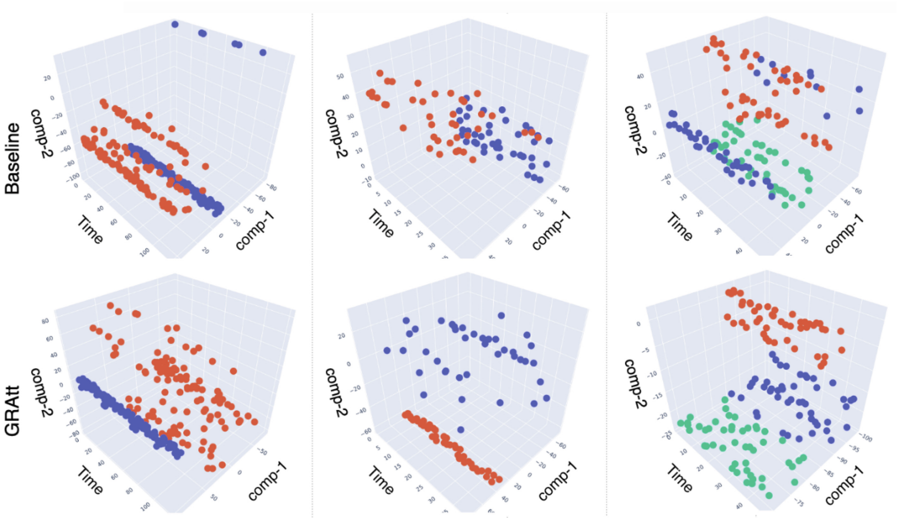


Fig. 4. Comparison of t-SNE embeddings. Each column depicts the t-SNE embeddings of predicted instances from the same video. Colors differentiate the instances. GRAtt produces temporally more consistent instance features essential for tracking.

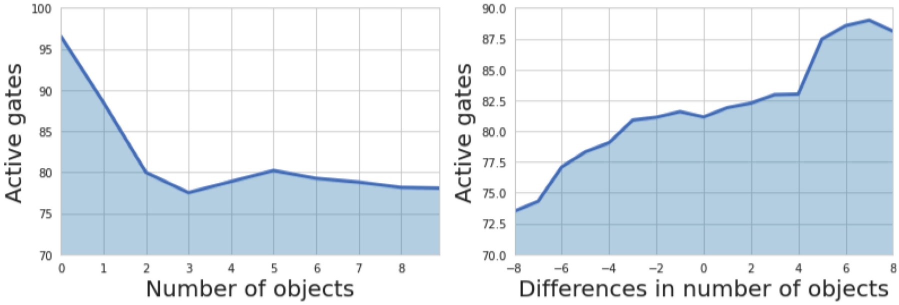


Fig. 5. Discrete Gate activation statistics on OVIS dataset w.r.t. object count and its difference between consecutive frames.

ing effectively retains past object queries in the presence of noise, ultimately benefiting the re-identification process. Figure 5 (right) shows gate activation w.r.t. the difference in the number of objects between frames. When the number of objects decreases, gate activation lowers, meaning the gate is preserving more from past frames. On the other hand, when the number of objects increases, the gate activation rises, thus selecting more features from the new or reappearing objects.

5 Qualitative Analysis

We illustrate the predictive ability of GRAtt-VIS on the OVIS dataset in Fig. 6. These videos contain diverse situations, such as slight occlusion (1st row), a rapidly moving person (2nd row), ducks with similar appearances crossing each other's paths (3rd row), and severe occlusion (4th row). We visualize more predictions of our model in the supplementary Fig. 9 and Fig. 10. To gain deeper insights into our model's operational principles, we have visualized the output of the gating mechanism in Fig. 3 and 6. Figure 3 illustrates the gate activations for a single video. We notice a decrease in active gates during occlusion or abrupt changes in the video. For example, gate activation is significantly reduced when the two horses start to occlude each other or when one of them vanishes. Similarly, the count drops again when the smaller horse goes through the severe occlusion. The last frames do not possess any complicated dynamics, and we

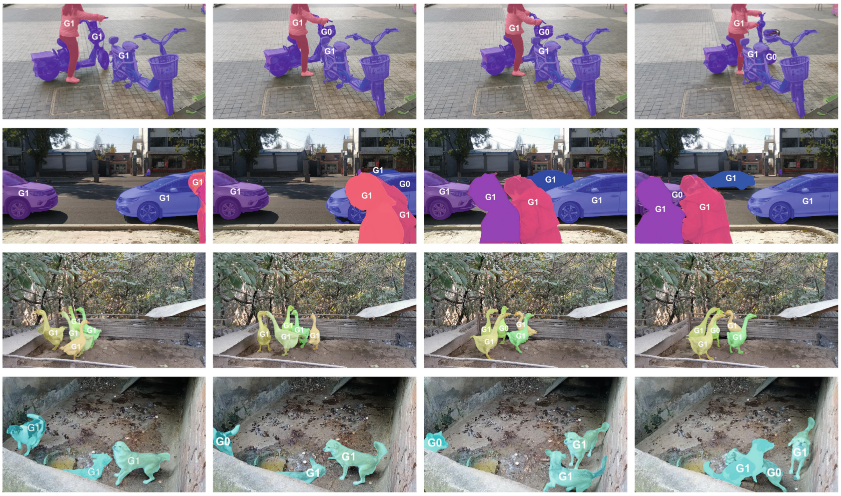


Fig. 6. Qualitative analysis of GRAtt-VIS on OVIS dataset. The overall and instance-specific gate activation is presented in each video. Notably, with the presence of occlusion and abrupt changes, the Gate activation shows a downward spike. For example, In the second video, we can see that the gate activation goes down significantly when the cars get occluded by the pedestrians.

observe an increasing trend of gate activation in that region of time. This highlights the model's ability to suppress degraded queries from the current frame during sudden changes. The gating signal in Fig. 6 turns to '0' when subjected to occlusion, preserving the non-occluded features from past frames.

6 Conclusion

In this work, we have introduced the GRAtt-VIS, a novel query propagation VIS method capable of absorbing shock in video frames and recovering degraded instance features. A gating mechanism is learned purely based on data without any heuristic to find an optimum discrete decision system for the propagation method. It can be seamlessly integrated into the family of query propagation-based methods. It simplifies the VIS pipeline by eliminating the need for a memory bank and, at the same time, performing on par state-of-the-art. We hope our work will accelerate research towards adaptive modeling of real-world dynamic scenarios.

References

1. Bertasius, G., et al.: Classifying, segmenting, and tracking object instances in video with mask propagation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9739–9748 (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
3. Cheng, B., Choudhuri, A., Misra, I., Kirillov, A., Girdhar, R., Schwing, A.G.: Mask2former for video instance segmentation. arXiv preprint arXiv:2112.10764 (2021)
4. Han, S.H., et al.: Visolo: grid-based space-time aggregation for efficient online video instance segmentation. In: CVPR (2022)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
6. Heo, M., et al.: A generalized framework for video instance segmentation. arXiv preprint arXiv:2211.08834 (2022)
7. Heo, M., Hwang, S., Oh, S.W., Lee, J.Y., Kim, S.J.: Vita: video instance segmentation via object token association. In: NeurIPS (2022)
8. Huang, D.A., Yu, Z., Anandkumar, A.: Minvis: a minimal video instance segmentation framework without video-based training. In: NeurIPS (2022)
9. Hwang, S., Heo, M., Oh, S.W., Kim, S.J.: Video instance segmentation using inter-frame communication transformers. In: NeurIPS (2021)
10. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
11. Koner, R., et al.: Instanceformer: an online video instance segmentation framework. arXiv preprint arXiv:2208.10547 (2022)
12. Li, M., Li, S., Li, L., Zhang, L.: Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In: CVPR (2021)
13. Lin, H., Wu, R., Liu, S., Lu, J., Jia, J.: Video instance segmentation with a propose-reduce paradigm. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1739–1748 (2021)

14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
15. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: a continuous relaxation of discrete random variables. CoRR abs/1611.00712 (2016). http://arxiv.org/abs/1611.00712
16. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: multi-object tracking with transformers. In: CVPR (2022)
17. Qi, J., et al.: Occluded video instance segmentation. arXiv preprint arXiv:2102.01558 (2021)
18. Thawakar, O., et al.: Video instance segmentation via multi-scale spatio-temporal split attention transformer (2022)
19. Voigtlaender, P., et al.: MOTS: multi-object tracking and segmentation. CoRR abs/1902.03604 (2019). http://arxiv.org/abs/1902.03604
20. Wang, Y., et al.: End-to-end video instance segmentation with transformers. In: CVPR (2020)
21. Wu, J., et al.: Efficient video instance segmentation via tracklet query and proposal. In: CVPR (2022)
22. Wu, J., Jiang, Y., Zhang, W., Bai, X., Bai, S.: Seqformer: a frustratingly simple model for video instance segmentation. In: ECCV (2022)
23. Wu, J., Liu, Q., Jiang, Y., Bai, S., Yuille, A., Bai, X.: In defense of online models for video instance segmentation. In: ECCV (2022)
24. Xu, N., et al.: Youtube-vis dataset 2021 version (2021). https://youtube-vos.org/dataset/vi. Accessed 01 Jan 2022
25. Xu, N., et al.: The 4th large-scale video object segmentation challenge - youtube-vos (2022). https://youtube-vos.org/challenge/2022/. Accessed 18 Aug 2022
26. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
27. Yang, S., et al.: Crossover learning for fast online video instance segmentation. In: ICCV (2021)
28. Yang, S., et al.: Tracking instances as queries. arXiv preprint arXiv:2106.11963 (2021)
29. Yang, S., et al.: Temporally efficient vision transformer for video instance segmentation. In: CVPR (2022)
30. Zhan, Z., McKee, D., Lazebnik, S.: Robust online video instance segmentation with track queries. arXiv preprint arXiv:2211.09108 (2022)
31. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: deformable transformers for end-to-end object detection. In: ICLR (2021)

AFMA-Track: Adaptive Fusion of Motion and Appearance for Robust Multi-object Tracking

Wei Liao[1], Lei Luo[2](✉), and Chunyuan Zhang[1]

[1] College of Computer Science and Technology, National University of Defence Technology, Changsha, China
{liaowei_v,cyzhang}@nudt.edu.cn

[2] Science and Technology on Parallel and Distributed Processing Laboratory,
College of Computer Science and Technology,
National University of Defense Technology, Changsha, China
l.luo@nudt.edu.cn

Abstract. Motion and appearance cues play a crucial role in Multi-object Tracking (MOT) algorithms for associating objects across consecutive frames. While most MOT methods prioritize accurate motion modeling and distinctive appearance representations, the use of appearance and motion cues is often confined to simplistic association techniques. For instance, fixed weights are commonly employed to combine the intersection-over-union (IoU) matrix and appearance similarity matrix, yielding an association cost matrix. To harness the full potential of motion and appearance cues across diverse scenarios, we propose an innovative approach that dynamically balances motion and appearance cues based on scene and object information during the association process. Furthermore, we introduce a new mechanism for updating appearance representations, effectively mitigating noise introduced by occlusion. Our method demonstrates state-of-the-art performance on the MOT17 and MOT20 test sets.

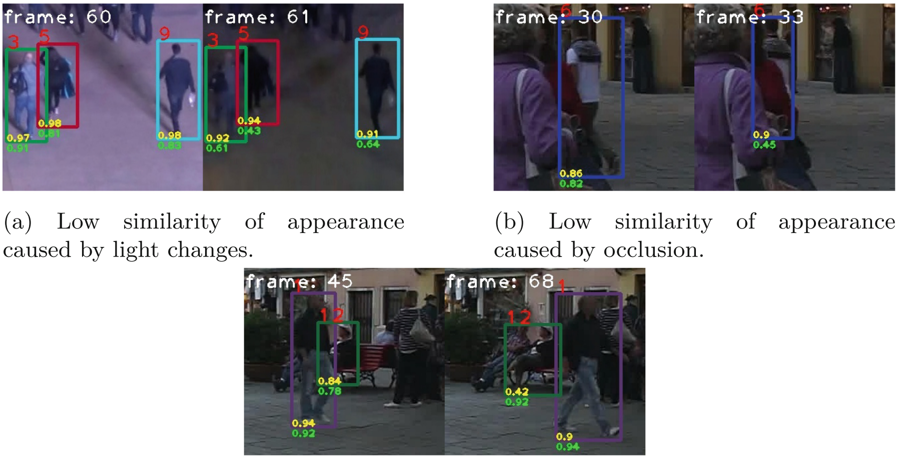
Keywords: MOT · tracking-by-detection · motion · appearance

1 Introduction

Multi-object tracking (MOT) is a fundamental task in computer vision and has a wide range of applications in many downstream tasks, such as autonomous driving, video surveillance, robot navigation. It aims to detect all interested objects in a video stream and to track the trajectory of all objects.

In general, MOT algorithms can be classified into tracking-by-detection (TbD) and tracking-by-regression paradigms, depending on whether data association is included or not. TbD is the prevalent paradigm for MOT, which comprises three phases: (1) object detection in the current frame, (2) prediction of tracklets positions with the extraction of pertinent cues (such as appearance

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 283–298, 2025.
https://doi.org/10.1007/978-3-031-78444-6_19



(a) Low similarity of appearance caused by light changes.

(b) Low similarity of appearance caused by occlusion.

(c) Low IoU caused by short-term occlusion.

Fig. 1. Examples of low appearance similarity or low IoU (motion). The yellow and green numbers represent the IoU and appearance similarity between the tracklet and the corresponding detection, respectively (Color figure online)

features) and (3) the association of detections and tracklets using these cues. The association phase is pivotal and relies on two key cues: object motion and appearance. The former is used to predict the position of the tracklets and then calculate the IoU matrix between the tracklets and the detections. Simultaneously, the latter is utilized to measure the similarity in appearance between the tracklets and the detections.

Recently, the majority of TbD methods [1–6] have focused on optimizing either motion or appearance models to obtain more precise location information or more distinctive appearance representations. However, the use of appearance and motion cues is often confined to simplistic association techniques. For instance, fixed weights are commonly employed to combine the IoU matrix and appearance similarity matrix, yielding the result association cost matrix. During the application of these TbD methods, we observed the following issues: (i) As illustrated in Fig. 1a, the IoU between tracklet prediction and detection remains relatively stable even in the case of sudden changes in scene illumination, while the appearance similarity drops significantly. (ii) As depicted in Fig. 1b, appearance similarity decreases notably when the object is occluded, and the object's appearance representation contains considerable noise at this point. These observations clearly indicates that employing a fixed weight to merge motion and appearance cues across all scenes is not justified. As shown in Fig. 1c, although appearance cues may not be very stable, they play an indispensable role in long-term association. It's a better way to assign different weights to motion cues and appearance cues based on different scene conditions. For dark scenes and crowded objects, we should place more trust in motion cues, while for bright scenes and substantial differences in appearance between objects, we should increase the weight of appearance cues.

In this paper, we propose two strategies to address the identified issues. Firstly, we introduce an adaptive fusion module that dynamically determines fusion weights between IoU and appearance similarity for each frame. This approach leverages scene-specific and object-related information to automatically select the appropriate fusion weights for each scene, eliminating the need for manual selection. Additionally, we propose a new technique for updating appearance that takes into account the degree of object occlusion. This technique aims to mitigate the impact of noise features from occluded objects, thereby enhancing the quality of appearance features.

By combining our methods into Bot-SORT [7], we achieve state-of-the-art results on the MOT17 [8] and MOT20 [9] test sets. To summary, our contribution are as follows:

- We present a method for assessing the quality of motion (IoU) and appearance cues for tracking, and accordingly, develop an adaptive fusion module to combine IoU and appearance similarity matrices to address variations quality of them across different scenarios.
- We further propose a dynamic appearance update technique that utilizes the IoU matrix extracted from the adaptive fusion module to calculate the occlusion degree of each object, aiming to mitigate the impact of noise features from occluded objects.

2 Related Work

Tracking-by-Detection. Due to advancements in detectors, contemporary methods predominantly adhere to the TbD paradigm. These approaches leverage detectors to identify objects within video sequences and subsequently utilize identity information for object association. Motion information serves as a foundational aspect for object association, with methods like SORT [10] employing Kalman filtering [11] to predict future object positions based on the assumption of constant motion, followed by employing the Hungarian algorithm [12] for association. Recent efforts [2,13,14] have integrated neural networks to enhance position prediction accuracy, constructing motion models for more intricate motion prediction. For instance, TrajE [13] models object motion using a Gaussian mixture model and utilizes Gated Recurrent Unit (GRU) [15] to predict parameters within the mixture. Meanwhile, ArTIST [14] represents motion as a discrete probability distribution to better capture natural pedestrian movement, additionally considering pedestrian interaction and integrating temporal information using GRU. However, relying solely on motion information may not effectively address object occlusion. Consequently, DeepSort [16] incorporates appearance information into SORT to enhance association accuracy. Both motion and appearance cues are crucial for association, and their judicious utilization can significantly improve association effectiveness. Our method aims to effectively incorporate both motion and appearance cues by analyzing environmental and object-related information.

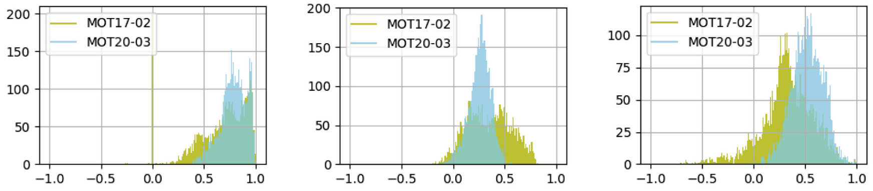
Data Association. Data association constitutes a crucial step in the TbD paradigm. In the early TbD methods, such as in SORT, association relied solely on IoU. DeepSort innovatively incorporates the appearance similarity of objects into the association process, enhancing its performance. MOTDT [17] generates candidate objects from detection and tracking objects and then uses a new scoring mechanism to select the final candidates. FairMOT [18] combines the Mahalanobis distance with the cosine distance calculated on re-identification (re-ID) features. ByteTrack [19] introduces the idea that detection objects with low confidence should not be discarded during association, contributing to a more robust tracking system. While numerous algorithms [5,18,19] conventionally employ a fixed weight to amalgamate IoU and appearance similarity during the association process, Deep OC-SORT [20] adopts varying fusion weights for distinct objects, albeit relying on predefined fixed weights for adaptive fusion. In contrast, our proposed method integrates both object-specific information and scene features to compute fusion weights at the image level. This approach ensures a balanced utilization of motion and appearance cues without necessitating manual weight selection. Besides, prevalent global association algorithms such as the Hungarian algorithm are commonly utilized during the association phase, our method, which determines fusion weights per image, aligns more effectively with these algorithms than the approach of calculating varied fusion weights for individual objects in Deep OC-SORT.

Appearance Update Strategies. Appearance cues are vital for distinguishing and re-identifying objects in tracking tasks. Most of the methods [4,7,19] use the Exponential Moving Average (EMA) mechanism to fuse the appearance features of the current frame with the historical appearance features to obtain more robust appearance features. Deep OC-SORT [20] argues that an object's detection confidence serves as a reliable indicator of its appearance feature quality. Fusion weights are consequently determined based on this confidence metric. Moreover, in our evaluation of appearance feature quality, we consider not only detection confidence but also factor in the degree of object occlusion. This comprehensive assessment enhances understanding of appearance feature reliability in diverse tracking scenarios.

3 The Quality of Motion and Appearance Cues for Tracking

We use Ground Truth to evaluate the quality of motion and appearance cues for tracking in different frames. For tracklet i, let p_i^a, p_i^m as the similarity of appearance and IoU with the corresponding detection under the current frame respectively, we define the quality of appearance cues r_i^a and the quality of motion cues r_i^m as follow:

$$r_i^a = p_i^a - \max_{j \neq i}(A_{ij}^T), r_i^m = p_i^m - \max_{j \neq i}(I_{ij}^T), \tag{1}$$

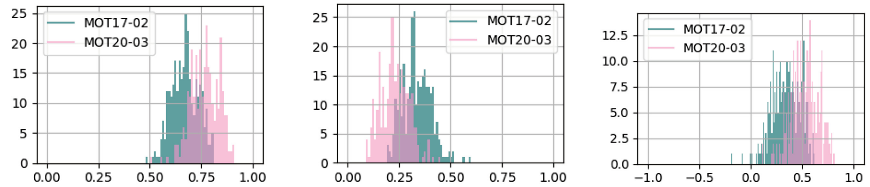


(a) Motion.

(b) Appearance.

(c) Disparity.

Fig. 2. The quality of motion and appearance cues varies across different scenarios, (c) is the statistical histogram of motion cues quality minus appearance cues quality



(a) Motion.

(b) Appearance.

(c) Disparity.

Fig. 3. Fusion weights for motion cues and appearance cues in different scenarios, (c) is the statistical histogram of motion cues weighting minus appearance cues weighting

where A_{ij}^T, I_{ij}^T represent the appearance similarity and IoU between the i-th tracklet and the j-th detection object.

We performed a statistical analysis on the MOT dataset using the aforementioned definition. Specifically, we analyzed 4000 samples each from MOT17-02 and MOT20-03. Figure 1a shows MOT20-03, while Fig. 1b depicts MOT17-02. These results are illustrated in Fig. 2. Despite the low illumination conditions in MOT17-02, MOT20-03 exhibits variations in lighting and smaller objects, suggesting a greater reliance on appearance cues in MOT17-02. However, the MOT17-02 sequence, characterized by a low camera angle, introduces significant occlusion, diminishing the impact of motion cues compared to the MOT20-03 scenario with a higher camera angle. After subtracting the quality of appearance cues from that of motion cues in both scenarios (Fig. 2c), it becomes evident that motion cues exert a more substantial influence than appearance cues in both scenarios. This influence is particularly pronounced in MOT20-03 compared to MOT17-02. Although motion cues generally exhibit higher quality compared to appearance cues, they tend to be less effective in addressing short-term occlusions. This is illustrated by the significant number of 0 values depicted in Fig. 2b. On the other hand, appearance cues demonstrate robustness when faced with short-term occlusions. Hence, adjusting fusion weights based on scenario-specific characteristics becomes imperative for optimizing tracking performance. For instance, assigning a higher weight to motion cues in MOT20-03 compared to MOT17-02.

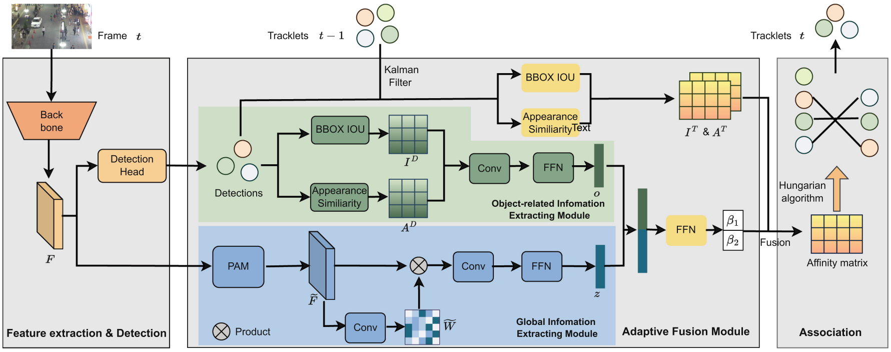


Fig. 4. An overview of AFMA-Track. (Conv: Convolution Layer, FFN: Feedforward Neural Network)

To validate the capability of our module in deriving fusion weights based on environmental and object-specific information. We randomly selected the fusion weights of 300 images from each of the MOT17-02 and MOT20-03 datasets for statistical purposes, as depicted in Fig. 3. The results indicate that the distribution of fusion weights aligns with the prior analyses of motion cues and appearance cues (Fig. 2). Specifically, the fusion weights related to motion cues surpass those related to appearance cues in both the MOT17-02 dataset and the MOT20-03 dataset. Moreover, the motion cues weights in MOT20-03 are notably higher than those in MOT17-02. These findings underscore the adaptive fusion module's ability to leverage environmental and object-related information to assess the quality of motion cues and appearance cues.

4 AFMA-Track

4.1 Overview

As depicted in Fig. 4, our approach adheres to the TbD paradigm. In the current frame t, we employ detector to acquire detections $D^t = \{d_i^t\}_{i=1}^M$, where M represents the number of detections. Simultaneously, we retain the feature map F extracted from the backbone. Subsequently, we compute the IoU matrix $I^D \in \mathbb{R}^{M\times M}$ and the appearance similarity matrix $A^D \in \mathbb{R}^{M\times M}$ based on the detections in the current frame. Next, our adaptive fusion module extracts global information from F and object-related information from I^D and A^D, yielding fusion weights β_1 and β_2. During the association phase, we utilize the Kalman Filter (KF) to predict the position of tracklets, computing the IoU matrix $I^T \in \mathbb{R}^{N\times M}$ and the appearance similarity matrix $A^T \in \mathbb{R}^{N\times M}$ between tracklets and detections, where N denotes the number of tracklets. The fusion weights β_1 and β_2 are applied to combine I^T and A^T into the cost matrix C, followed by the utilization of the Hungarian algorithm for matching. Post-matching, we implement a dynamic appearance update strategy to refine the appearance features of the tracklets.

4.2 Adaptive Fusion Module

To effectively allocate fusion weights for appearance and motion cues in different scenarios, our adaptive fusion module extracts scene information and object-related information to determine the fusion weights of the IoU matrix I^T and the appearance similarity matrix A^T.

Global Information Extracting Module. We utilize the backbone to extract the feature map F from the image. Subsequently, the Position Attention Module (PAM) [21] is employed to obtain preliminary global context information. PAM facilitates the capture of spatial dependencies across the feature map by introducing an attention mechanism. Specifically, the feature at each position is updated through a weighted summation of features from all positions, with weights determined by the feature similarity between the corresponding positions. This mechanism enhances the model's ability to capture intricate spatial relationships within the feature map.

We derive the final global information from the feature maps $\widetilde{F}$ processed by the PAM, employing a method akin to the approach described in [22]. This process can be represented as:

$$\begin{aligned} \widetilde{W} &= \text{Softmax}(\text{conv}_{1\times1}(\widetilde{F})), \\ z &= \text{FFN}(\text{conv}_{3\times3}(\sum_{i=1}^{H}\sum_{j=1}^{W}\widetilde{W}_{i,j}\widetilde{F}_{i,j})), \end{aligned} \tag{2}$$

where $\widetilde{W}$ is a weight map extracted from $\widetilde{F}$, H and W represent the dimensions of the feature map, and z represents the global context vector.

Object-Related Information Extracting Module. We dissect the informational interplay between objects into two pivotal dimensions: location interaction and appearance similarity. The matrix I^D meticulously delineates the spatial relationships among objects, capturing their intricate interactions. Simultaneously, the matrix A^D is employed to encode the semblances in appearance, providing insights into the visual similarities between objects. To accommodate scenarios with different numbers of objects, we expand I^D and A^D to a fixed dimension $M_f \times M_f$, and the value of M_f is manually determined.

We stitch I^D and A^D together and use a 1×1 convolution to obtain matrix O_1. As each row within matrix O_1 encapsulates the relationships between an individual object and others, we subsequently apply a horizontal convolution to enhance information extraction:

$$\begin{aligned} O_1 &= \delta(\text{BN}(\text{conv}_{1\times1}(\text{Cat}(I^D, A^D)))), \\ O_2 &= \delta(\text{BN}(\text{conv}_{1\times N}(O_1)), \end{aligned} \tag{3}$$

where δ denotes the ReLU, BN is Batch Normalization. Afterwards, we use a 1×1 convolution and a FFN to process F_2 to obtain the final object-related information vector o.

After obtaining the global context vector and the object-related information vector, we concatenate the two vectors and then process them to calculate the final fusion weights:

$$\beta_1, \beta_2 = \text{Softmax}(\text{FFN}(\text{Cat}(z, o))). \tag{4}$$

Training. The main objective of the matching process is to maximize the similarity between tracklets and their corresponding detections while minimizing the similarity between tracklets and detections with distinct IDs. However, considering all the similarities between tracklets and detections during training is unnecessary, what really affects the correlation is often the detection with the highest similarity beyond the detection corresponding to the tracklet. Therefore, it suffices to focus on the similarity between these two detections and the tracklet. Let p_i represent the similarity between tracklet i and its corresponding detection in the current frame. We use the focal loss function to train adaptive fusion module:

$$\mathcal{L} = \frac{1}{N} \sum_{i}^{N} (\text{FL}(p_i, \max_{j \neq i}(C_{ij}))), \tag{5}$$

where C_{ij} is the similarity between the i-th tracklet and the j-th detection, and FL represents the focal loss.

4.3 Dynamic Appearance Update Strategy

In previous work [4,7,19], the appearance embeddings of matched tracklets were updated using the EMA mechanism, which requires a fixed weight α to adjust the ratio between the current frame's embeddings and the historical appearance embeddings. Let e_i^t be the appearance state of the i-th tracklet at the t-th frame. The standard EMA is

$$e_i^t = \alpha e_i^{t-1} + (1 - \alpha) e_i, \tag{6}$$

where e_i is the appearance embedding of the matched detection, and α is usually set to 0.9.

However, the EMA mechanism overlooks the influence of noise features that arise due to occluded objects. To address this issue, we propose to adapt the value of α based on the degree of occlusion between objects. The IoU matrix effectively captures the occlusion between objects, and assessing the degree of occlusion involves identifying which objects are obstructed by others. Here, the detection confidence of the object reflects this relationship, with the detection confidence of the obstructed object typically being lower compared to that of the object causing the occlusion. To quantify the degree of occlusion of the i-th object, we define σ_i as follows:

$$\hat{I}^D_{i,j} = \begin{cases} I^D_{i,j}, & s_j \geq s_i \\ 0, & s_j < s_i, \end{cases} \tag{7}$$

$$\sigma_i = \begin{cases} \min\{\sum_{j=1,j\neq i}^{M} \hat{I}^D_{i,j}, 1\}, & s_i \geq s_t \\ 1, & s_i < s_t, \end{cases} \tag{8}$$

where s_i is the detector confidence of i-th detection, s_t is the detection confidence threshold and objects with detector confidence larger than this value are considered to be present in the frame, , $I^D_{i,j}$ is the IoU between the i-th detection and j-th detection, $\hat{I}^D$ is the filtered IoU matrix. As we employ the association method described in [19], objects with detection confidence below the threshold may also be associated. However, it is observed that many of these objects with low detection confidence exhibit significant occlusion or motion blur, leading to the presence of substantial noise in the acquired appearance features. Hence, we assign an occlusion degree of 1 to these objects.

With σ_i, we replace α with a new α_i defined as

$$\alpha_i = 1 - (1 - \alpha)(1 - \sigma_i), \tag{9}$$

we set the fixed value $\alpha = 0.95$ follow [20]. When $\sigma_i = 1$, we set $\alpha_i = 1$, resulting in the complete disregard of the new appearance embedding. On the other hand, if $\sigma_i = 0$, then $\alpha_i = \alpha$, leading to the maximal contribution of e_i to the update of the tracklet appearance embedding.

5 Experiments

5.1 Settings

Datasets. To validate the effectiveness of our method, we conduct experiments on the MOT17 and MOT20 datasets under the "private detection" protocol. The MOT17 dataset encompasses diverse scenes, featuring various camera movements and angles, while MOT20 emphasizes crowded and complex environments, including indoor and outdoor settings, as well as scenarios with varying light conditions. Both MOT17 and MOT20 exclusively consist of training and test sets, without a separate validation set. We follow [7,19,30], when doing ablation experiments, the first half of each video sequence in the training set is used for training and the second half is used for validation.

Metrics. We apply CLEAR metrics [31] which includes HOTA [32], AssA, DetA, MOTA and IDF1 [33], etc. to evaluate different aspects of the tracking performance. HOTA is currently the main metric used to evaluate tracking performance, taking into account a balance of detection accuracy, matching accuracy

Table 1. Comparsion of the state-of-the-art methods under the "private detector" protocol on MOT17 test set. **Bold** represents the best results and underlining for the second best result

Tracker	HOTA↑	AssA↑	AssR↑	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
FairMOT [18]	59.3	58.0	63.6	72.3	73.7	27507	117477	3303
CSTrack [23]	59.3	57.9	63.2	72.6	74.9	23847	114303	3567
TransCenter [24]	54.5	49.7	54.2	62.2	73.2	23112	123738	4614
TransTrack [25]	54.1	47.9	57.1	63.5	75.2	50157	86442	4872
MOTR [26]	62.0	60.6	65.6	75.0	78.6	23409	94797	2619
RelationTrack [22]	61.0	61.5	67.3	74.7	73.8	27999	118623	2166
MotionTrack [2]	65.1	65.1	70.8	80.1	81.1	23802	81660	1140
FineTrack [4]	64.3	64.5	70.1	79.5	80.0	217500	90096	1272
OC-SORT [1]	63.2	63.4	67.5	77.5	78.0	**15129**	107055	1950
StrongSORT [27]	64.4	64.4	70.0	79.5	79.6	27876	86205	1866
ByteTrack [19]	63.1	62.0	68.2	77.3	80.3	25491	83721	2196
UTM [28]	64.0	62.5	69.1	78.7	**81.8**	25077	**76298**	1431
BPMTrack [29]	63.6	62.0	68.4	78.1	81.3	25785	77859	2010
BoT-SORT [7]	65.0	65.5	71.2	80.2	80.5	22521	86037	1212
AFMA-Track (ours)	**65.31**	**65.78**	**71.79**	**80.66**	80.66	23634	84363	**1113**

and positioning accuracy. AssA is the metric that evaluates association accuracy and AssR is to evaluate the recall of the association. MOTA emphasizes detection performance and IDF1 focuses on identity association performance.

Implementation Details. To ensure a fair comparison, we employ the YOLOX model trained by [19] for object detection and the FastReID's SBS-50 model trained by [7] for extracting appearance features. For MOT17, we set $M_f = 128$, while for MOT20, we use $M_f = 256$. In the linear assignment stage, we reject a match if the IoU is less than 0.2 when considering only IoU. Similarly, if only appearance similarity is considered, a match is rejected if the appearance similarity is less than 0.6. When both are used, we adjust the rejection thresholds according to the fusion weight of IoU and appearance similarity in the AFMA-Track. Additionally, to prevent premature termination, we retain lost tracklets for 30 frames. We have also adapted the Modified KF (MKF) algorithm by setting the object's bounding box size and speed at the time of loss to the average values obtained from previous frames before the loss occurred. During the lost period, we ensure consistency in the object's bounding box size follow [7]. For other parameter settings, we maintain consistency with BoT-SORT.

During the training phase, we implemented a random successive sampling strategy, selecting 8 consecutive frames randomly from the video sequence for training. Specifically for MOT17, we initially trained for 4 epochs using this strategy, followed by an additional 4 epochs with sequential sampling on the MOT17 training set, totaling approximately 8 h. For MOT20, a similar approach was

Table 2. Comparsion of the state-of-the-art methods under the "private detector" protocol on MOT20 test set

Tracker	HOTA↑	AssA↑	AssR↑	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
Decode-MOT [34]	54.5	54.6	58.4	69.0	67.2	35217	131502	2805
FairMOT [18]	54.6	54.7	57.7	67.3	61.8	103440	88901	5243
CSTrack [23]	54.0	54.0	57.6	68.6	66.6	25404	144358	3196
MAA [35]	57.3	55.1	61.1	71.2	73.9	24942	108744	1331
TransCenter [24]	43.5	37.0	45.1	49.6	58.5	64217	146019	4695
TransTrack [25]	48.9	45.2	51.9	59.4	65.0	27191	150197	11352
ReMOT [36]	61.2	58.7	63.1	73.1	77.4	28351	86659	2121
QDTrack [37]	60.0	58.9	65.7	73.8	74.7	23352	106313	1042
MotionTrack [2]	62.8	61.8	68.0	76.5	**78.0**	28629	**84152**	1165
StrongSORT [27]	62.6	**64.0**	**69.6**	77.0	73.8	**16632**	117920	**1003**
OC-SORT [1]	62.4	62.5	67.4	76.3	75.7	19067	105894	1086
ByteTrack [19]	61.3	59.6	66.2	75.2	77.8	26249	87594	1223
BoT-SORT [7]	63.3	62.9	68.6	77.5	77.8	24638	88863	1313
AFMA-Track (ours)	**63.47**	63.14	69.16	**77.76**	77.81	26222	87385	1207

taken, with 4 epochs of training using random successive sampling followed by 2 epochs using sequential sampling on the MOT20 dataset, requiring around 12 h. The training was conducted on an NVIDIA GeForce RTX 2080ti GPU with a batch size of 1. We employ the SGD optimizer with a weight decay of 5×10^{-4} and momentum of 0.9. The initial learning rate is set to 10^{-4} with a 1 epoch warmup and cosine annealing schedule.

5.2 Benchmark Results

We compare our method with the state-of-the-art trackers on the test of MOT17 and MOT20 under the private detection protocol.

MOT17. As show in Table 1, our method outperforms the state-of-the-art methods on many key metrics. i.e. rank first for metrics in HOTA, IDs, IDF1, AssA and AssR . By focusing on enhancing object association accuracy, we have successfully achieved exceptional performance in diverse scenarios. The notable high scores in HOTA and IDF1 further underline our ability to achieve robust and precise object associations.

MOT20. Our method exhibits strong performance on the intricate and densely populated MOT20 dataset. Table 2 illustrates our results in comparison to state-of-the-art methods, highlighting our method's leading positions in terms of IDF1 and HOTA, as well as second place rankings in AssA, AssR, and MOTA. These results emphasize the exceptional effectiveness and robustness of our approach. Despite our method's strong performance, it is noteworthy that StrongSORT [27] outperforms our approach in metrics related to the association effect, specifically

Table 3. Ablation experiments on the MOT17 validation set. (MKF: Modified Kalman Filter, AW: Adaptive Weighting, DA: Dynamic Appearance)

Setting	HOTA↑	AssA↑	AssR↑	IDF1↑	MOTA↑	IDs↓	FPS↓
Baseline (BoT-SORT)	68.802	70.700	76.081	81.575	78.601	456	**6.335**
Baseline+AW	69.553	72.254	77.503	82.141	**78.692**	**442**	6.188
Baseline+AW+MKF	69.641	72.422	77.603	82.504	**78.692**	444	6.187
Baseline+AW+MKF+DA	**69.947**	**73.028**	**77.768**	**82.944**	78.688	445	6.172

Table 4. Ablation study of various association strategies on BoT-SORT

Fusion Strategy	HOTA↑	AssA↑	AssR↑	IDF1↑	MOTA↑	IDs↓
Fairmot Strategy [18]	67.973	69.484	74.822	79.767	77.641	527
BoT-SORT Original [7]	68.802	70.700	76.081	81.575	78.601	456
Deep OC-SORT Strategy [20]	69.033	71.229	76.013	81.284	78.397	491
AW (ours)	**69.553**	**72.254**	**77.603**	**82.141**	**78.692**	**442**

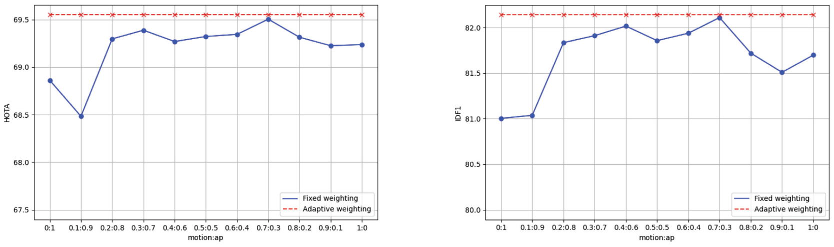
AssA and AssR. This superiority can be attributed to StrongSORT's utilization of an offline global association method, contrasting with our online association approach. Additionally, because our method optimizes the association process by leveraging acquired motion and appearance cues through adaptive weight fusion, its performance is closely tied to the quality of these cues, and the improvement in performance is also constrained by the quality of these two cues.

5.3 Ablation Study

Effect of Each Component. We conduct ablation experiments on the MOT17 validation set to evaluate the effectiveness of each component in our method. To ensure a fair comparison, we keep all other baseline settings consistent. As presented in Table 3, the adaptive fusion module exhibits significant improvements in HOTA, AssA, AssR, and IDF1, indicating the efficacy of adaptive fusion weights. We introduce the MKF as a simple modification to KF, which mitigates error accumulation caused by occlusions to some extent. The observed enhancements in HOTA and IDF1 further validate this approach. Additionally, the dynamic appearance update strategy effectively boosts HOTA and IDF1, proving its effectiveness for correlation. It is noteworthy that our methods have minimal impact on MOTA, as it is primarily influenced by detection performance, while our optimizations primarily focus on the association aspect.

Computational Overhead. As shown in Table 3, our method introduces additional modules, incurring some computational overhead. However, the impact on processing speed is minimal, with a decrease of less than 0.2 frames per second (FPS). The incurred overhead is nearly negligible, highlighting the efficiency of our approach.

Analysis of Adaptive Weighting. In order to validate the advantages of our adaptive fusion module, we compare it with several other fusion strategies on BoT-SORT while maintaining other settings consistent. As depicted in Table 4, our method outperforms the fusion strategies of FairMOT and BoT-SORT across several metrics such as HOTA, AssA, AssR, and more. Despite Deep OC-SORT employing different fusion weights for different objects, our method achieves a 0.5% higher HOTA and 0.8% higher IDF1 compared to it, fully demonstrating the superiority of the adaptive fusion module and the significance of using different fusion weights in varying scenarios. Additionally, as illustrated in Fig. 5, we showcase the impact of utilizing different fixed fusion weights; IoU alone can already yield satisfactory results, while a 0.7:0.3 fusion weighting strategy produces even better results. However, our adaptive module attains the best results without the need for manual selection of fusion weights.



(a) HOTA.

(b) IDF1.

Fig. 5. Performance with different fixed fusion weights on BoT-SORT

Analysis of Dynamic Appearance. To demonstrate the superiority of our dynamic appearance update strategy, we compare it with three other update strategies on the basis of AW and MKF-enhanced BoT-SORT, keeping all other settings consistent. The results, as presented in Table 5, show that the dynamic

Table 5. Ablation Experiments for Different Appearance Update Strategies on AW and MKF-enhanced BoT-SORT. "-" represents using the currently detected appearance embedding as the appearance embedding of the tracklet, and AP is average appearance embedding update strategy

Settings	HOTA↑	AssA↑	AssR↑	IDF1↑	MOTA↑	IDs↓
–	69.189	71.530	76.977	81.918	78.566	449
EMA	69.641	72.422	77.603	82.504	**78.692**	**444**
AP	69.537	72.271	77.335	82.669	78.675	450
Deep OC-SORT Strategy [20]	69.773	72.703	77.727	82.879	78.631	448
DA (ours)	**69.947**	**73.028**	**77.768**	**82.944**	78.688	445

appearance update strategy significantly outperforms the other strategies in key metrics such as HOTA and IDF1, while also being comparable in terms of MOTA and IDs. While Deep OC-SORT evaluates and diminishes the noise impact on appearance features based on the object's detection confidence, our dynamic appearance update strategy further effectively mitigates noise interference arising from occlusion by assessing the degree of occlusion for each object.

6 Conclusion

In this paper, we propose a new TbD method to address the varying impact of motion and appearance cues across different scenarios. By designing an adaptive fusion module to obtain the fusion weights of IoU and appearance similarity matrices based on scene information and object-related information. And we additionally propose a dynamic appearance update strategy to reduce the impact of noise features from occluded objects. We validate the effectiveness of each component, and our results on MOT Benchmark can demonstrate the benefits of our method.

Acknowledgements. This research was supported by PDL foundation of No. WDZC20235250116 and NSFC of No. 61872377.

References

1. Cao, J., Pang, J., Weng, X.: Observation-centric sort: rethinking sort for robust multi-object tracking. In: CVPR, pp. 9686–9696 (2023)
2. Qin, Z., Zhou, S., Wang, L.: MotionTrack: learning robust short-term and long-term motions for multi-object tracking. In: CVPR, pp. 17939–17948 (2023)
3. Ma, C., Yang, C., et al.: Trajectory factory: tracklet cleaving and re-connection by deep siamese Bi-GRU for multiple object tracking. In: IEEE ICME, pp. 1–6 (2018)
4. Ren, H., Han, S., Ding, H.: Focus on details: online multi-object tracking with diverse fine-grained representation. In: CVPR, pp. 11289–11298 (2023)
5. Seidenschwarz, J., Brasó, G., Serrano, V.C.: Simple cues lead to a strong multi-object tracker. In: CVPR, pp. 13813–13823 (2023)
6. Yu, E., Li, Z., Han, S.: Towards discriminative representation: multi-view trajectory contrastive learning for online multi-object tracking. In: CVPR, pp. 8834–8843 (2022)
7. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: BoT-SORT: robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
8. Milan, A., Leal-Taixé, L., Reid, I.: MOT16: a benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
9. Dendorfer, P., Rezatofighi, H., Milan, A.: MOT20: a benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
10. Bewley, A., Ge, Z., Ott, L.: Simple online and realtime tracking. In: ICIP, pp. 3464–3468 (2016)

11. Kalman, R.E.: A new approach to linear filtering and prediction problem (1960)
12. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**(1–2), 83–97 (1955)
13. Girbau, A., Giró-i Nieto, X., Rius, I.: Multiple object tracking with mixture density networks for trajectory estimation. arXiv preprint arXiv:2106.10950 (2021)
14. Saleh, F., Aliakbarian, S., Rezatofighi, H.: Probabilistic tracklet scoring and inpainting for multiple object tracking. In: CVPR, pp. 14329–14339 (2021)
15. Cho, K., et al.: Learning phrase representations using RNN encoder–decoder for statistical machine translation, pp. 1724–1734 (2014)
16. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: ICIP, pp. 3645–3649 (2017)
17. Chen, L., Ai, H., et al.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: IEEE ICME, pp. 1–6 (2018)
18. Zhang, Y., Wang, C., Wang, X.: FairMOT: on the fairness of detection and re-identification in multiple object tracking. IJCV **129**, 3069–3087 (2021)
19. Zhang, Y., et al.: ByteTrack: multi-object tracking by associating every detection box. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13682, pp. 1–21. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20047-2_1
20. Maggiolino, G., Ahmad, A., Cao, J., Kitani, K.: Deep oc-sort: multi-pedestrian tracking by adaptive re-identification. In: ICIP, pp. 3025–3029. IEEE (2023)
21. Fu, J., Liu, J., Tian, H.: Dual attention network for scene segmentation. In: CVPR, pp. 3146–3154 (2019)
22. Yu, E., Li, Z., Han, S.: Relationtrack: relation-aware multiple object tracking with decoupled representation. IEEE TMM (2022)
23. Liang, C., Zhang, Z., Zhou, X., et al.: Rethinking the competition between detection and reid in multiobject tracking. IEEE TIP **31**, 3182–3196 (2022)
24. Xu, Y., Ban, Y., et al.: TransCenter: transformers with dense representations for multiple-object tracking. IEEE TPAMI **45**(6), 7820–7835 (2022)
25. Sun, P., Cao, J., Jiang, Y.: Transtrack: multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)
26. Zeng, F., Dong, B., Zhang, Y.: Motr: end-to-end multiple-object tracking with transformer. In: ECCV, pp. 659–675 (2022)
27. Du, Y., Zhao, Z., Song, Y.: Strongsort: make deepsort great again. IEEE TMM (2023)
28. You, S., Yao, H., Bao, B., Xu, C.: Utm: a unified multiple object tracking model with identity-aware feature enhancement. In: CVPR, pp. 21876–21886 (2023)
29. Gao, Y., Haojun, X., Li, J., Gao, X.: Bpmtrack: multi-object tracking with detection box application pattern mining. IEEE TIP **33**, 1508–1521 (2024)
30. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: ECCV, pp. 474–490 (2020)
31. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. **2008**, 1–10 (2008)
32. Luiten, J., Osep, A., Dendorfer, P.: Hota: a higher order metric for evaluating multi-object tracking. IJCV **129**, 548–578 (2021)
33. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin, L.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV, pp. 17–35 (2016)
34. Lee, S.-H., Park, D.-H., Bae, S.-H.: Decode-mot: how can we hurdle frames to go beyond tracking-by-detection? IEEE TIP (2023)

35. Stadler, D., Beyerer, J.: Modelling ambiguous assignments for multi-person tracking in crowds. In: WACV, pp. 133–142. IEEE (2022)
36. Yang, F., Chang, X., Sakti, S., Yang, W., Nakamura, S.: Remot: a model-agnostic refinement for multiple object tracking. Image Vis. Comput. **106**, 104091 (2021)
37. Fischer, T., et al.: Qdtrack: quasi-dense similarity learning for appearance-only multiple object tracking. IEEE TPAMI (2023)

Structure from Motion with Variational Bayesian Inference in Multi-resolution Networks

Teruya Aburayama and Norio Tagawa(✉)

Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan
tagawa@tmu.ac.jp
https://t-lab.fpark.tmu.ac.jp/tagawalab/home/Welcome.html

Abstract. In this study, we investigate a depth recovery method based on optical flow from two consecutive frames with relative motion between the object and the camera. Multi-resolution processing is suitable for high-density depth recovery that avoids aliasing and preserves discontinuities. Propagation of the recovery results from the low-resolution layer to the high-resolution layer is an important issue. In this paper, we propose a method based on variational Bayesian inference. By computing the posterior distributions of the depth and motion parameters at each layer using the mean-field approximation and converting them into the prior distribution of the upper layer, it is possible to propagate the depth and motion information simultaneously. The effectiveness of the proposed method was quantitatively evaluated using artificial images, and its practicality of the system was also confirmed qualitatively.

Keywords: Structure from motion · Depth recovery · Optical flow · Variational Bayesian inference · Multi-resolution processing

1 Introduction

In recent years, human perception and consciousness have been elucidated on the basis of the free energy principle, and it is claimed that they function through variational Bayesian inference (VB) realized by bidirectional connections between nerve cells [5,6,20]. On the other hand, deep neural networks have solved various problems that artificial intelligence tries to solve, including depth recovery from images [2,7,10,13,18]. These processes do not necessarily occur in the same way in the human brain, but these functions are considered necessary as a process of acquiring the prior probabilities required by VB through experience. Conversely, VB can be seen as a process of processing current observations based on a physical model, using the knowledge gained from such experience. Therefore, in this study, we focus on 3D perception from a sequence of 2D images

Supported by Tokyo Metropolitan University.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 299–317, 2025.
https://doi.org/10.1007/978-3-031-78444-6_20

[1,9,17,21], called "Structure from Motion (SfM)," and investigate a new method within the framework of VB [11,12].

As a more primitive task, we consider the estimation of dense depth maps and relative camera motion from two consecutive frames. For this reason, this study is based on using optical flow rather than feature point correspondence. The problems to be solved in this framework can be summarized as follows:

1. The gradient equation is a first-order approximation of the invariance of image brightness before and after camera movement. Therefore, if there is a large optical flow relative to the scale of the image texture, aliasing will occur, and therefore a small, incorrect optical flow will satisfy the gradient equation.
2. In small areas with uniform texture, optical flow cannot be uniquely determined. To avoid this, it is necessary to assume that the optical flow is constant in large areas with different brightness gradients, but this prevents the detection of detailed depth structures, especially discontinuities.
3. The degrees of freedom in depth that can be recovered increases as the number of pixels increases. For this reason, it is natural to define and use a semi-parametric model in SfM. In this case, the estimation accuracy of not only the depth but also the camera motion parameters decreases due to the Neyman-Scott problem [3].

To comprehensively solve the above problems, we construct a Bayesian network in the resolution direction, which represents a multi-resolution model of depth [14]. We define a transition model between the resolutions by varying the block size, which is assumed to be constant depth, in conjunction with the resolution. To improve estimation accuracy, smooth depth models based on ridge regression and/or total variation regularization are often used, which contribute to the recovery of depth discontinuities, but are computationally expensive. In contrast, our hierarchical model assumes smooth depth over a large area at low resolutions and is able to represent fine structure as resolution increases, naturally handling depth discontinuities in the process. Furthermore, a hierarchical estimation framework that sequentially recovers depths from lower resolutions allows warping of higher resolution images based on optical flows computed from lower resolution depths, thus eliminating the aliasing.

In our previous work, only depth information was propagated in the Bayesian network [14]. In this study, the posterior distributions of depth and motion parameters are computed using VB with the mean-field approximation in the low-resolution layer and propagated together to the high-resolution layer. We quantitatively confirm the effectiveness of the proposed method on artificial images, such as the suppression of aliasing made possible by hierarchical inference and the ability of the hierarchical depth model to recover discontinuities. We also qualitatively evaluate the applicability to real images.

2 Gradient Equation and Optical Flow for Perspective Projection

Assuming that the camera motion between the two frames is small enough, the optical flow $(v_x(x, y, t), v_y(x, y, t))$, which is a 2D velocity field, satisfies the following gradient equation at each pixel.

$$f_t(x, y, t) = -f_x(x, y, t)v_x(x, y, t) - f_y(x, y, t)v_y(x, y, t), \tag{1}$$

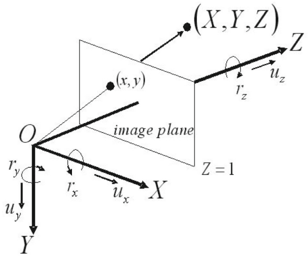


Fig. 1. Camera projection model and notation definition.

where f is the image brightness value, and f_x, f_y, and f_t are the spatiotemporal derivatives. In the gradient equation, f_x, f_y, and f_t are observations. In the gradient equation, an explicit observation equation can be specifically defined as follows.

$$f_t = -\bar{f}_x v_x - \bar{f}_y v_y + \epsilon_t, \quad f_x = \bar{f}_x + \epsilon_x, \quad f_y = \bar{f}_y + \epsilon_y. \tag{2}$$

In SfM assuming perspective projection, the optical flow for the rigid motion is formulated as follows:

$$v_x = xyr_x - (1 + x^2)r_y + yr_z - (u_x - xu_z)d \equiv v_x^r(\boldsymbol{r}) + v_x^u(\boldsymbol{u})d, \tag{3}$$

$$v_y = (1 + y^2)r_x - xyr_y - xr_z - (u_y - yu_z)d \equiv v_y^r(\boldsymbol{r}) + v_y^u(\boldsymbol{u})d, \tag{4}$$

where $\boldsymbol{r} = [r_x, r_y, r_z]^\top$ and $\boldsymbol{u} = [u_x, u_y, u_z]^\top$ indicate the rotational velocity vector and the translational velocity vector respectively and d corresponds to the shallowness $1/Z$. Figure 1 shows the camera projection system and the notations used in the text.

By accurately computing f_x and f_y with a spatial filter using the brightness values of the surrounding pixels [4], we assume that the observation errors ϵ_x and ϵ_y are sufficiently small. The errors associated with ϵ_x and ϵ_y on the right-hand side of Eq. 1 are then approximately incorporated into the error in f_t and are often treated as random errors. Even if we consider f_x and f_y as noise-free observations in this way, the unknown degrees of freedom for depth are large, and to reduce them, this study uses a hierarchical depth model.

3 Key Features of Our Approach

3.1 Gradient Equation for Rigid Motion

In the rigid motion analysis, the camera motion and depth satisfying Eqs. 2, 3 and 4 are estimated. In the past, many studies have been conducted on a two-step method in which (v_x, v_y) is detected from Eq. 2 and then analyzed based on Eqs. 3 and 4 [1,15]. Since this method expands the solution space on the way, there is a problem from the viewpoint of optimality. In this study, we consider the problem of substituting Eqs. 3 and 4 into Eq. 2 to implicitly detect an optical flow that completely satisfies the rigid motion constraint.

$$f_t = -\left\{f_x v_x^r(\boldsymbol{r}) + f_y v_y^r(\boldsymbol{r})\right\} - \left\{f_x v_x^u(\boldsymbol{u}) + f_y v_y^u(\boldsymbol{u})\right\} d \equiv -\boldsymbol{f}_r^\top \boldsymbol{r} - d\boldsymbol{f}_u^\top \boldsymbol{u}, \quad (5)$$

$$\boldsymbol{f}_r = \begin{bmatrix} f_x xy + f_y(1+y^2) \\ -f_x(1+x^2) - f_y xy \\ f_x y - f_y x \end{bmatrix}, \quad \boldsymbol{f}_u = \left[-f_x, -f_y, f_x x + f_y y\right]^\top . \quad (6)$$

Equation 5 is treated as an observation equation with f_t as the observation. For that purpose, it is effective to use a differential kernel instead of a simple first-order difference to determine f_x and f_y. In this study, we use the five-tap kernel $[-0.108415, -0.280353, 0, 0.280353, 0.108415]$ is applied to all the resolution layers as a differentiator [4].

When a rigid body can be assumed, depth and camera motion can be recovered directly from the image brightness by not treating optical flow explicitly. The optical flow is determined a posteriori from the estimated depth and motion parameters by Eqs. 3 and 4.

3.2 Multi-resolution Scheme to Suppress Aliases

The aliasing in this study is that when the optical flow is relatively large with respect to the spatial wavelength of the brightness pattern, the optical flow that is shorter than the actual length is detected as an artifact. The aliasing problem is effectively solved by multi-resolution processing. First, each of the two consecutive original images is decomposed into a multi-resolution image with an appropriate number of layers. From a low-resolution image pair, i.e., an image in which the wavelength of the brightness pattern is long, a low-resolution optical flow is detected that is less likely to cause aliases. Depth and motion parameters estimated at low resolution, along with their reliability, should be propagated accurately to the next higher resolution layer. This achieves estimation based on all brightness information while avoiding aliases. The proposal in this study is to implement such information propagation based on VB.

3.3 Time Derivative of Image Brightness Using Image Warping

In the multi-resolution process of this study, instead of calculating the depth perturbation corresponding to the resolution at each layer, the entire depth is updated. Therefore, the updated value of f_t in the gradient equation Eq. 5 must be calculated using image warping as follows.

$$f_t = -f_x \hat{v_x} - f_y \hat{v_y} + \frac{\partial}{\partial t}\mathcal{W}(f, \hat{\boldsymbol{v}}), \tag{7}$$

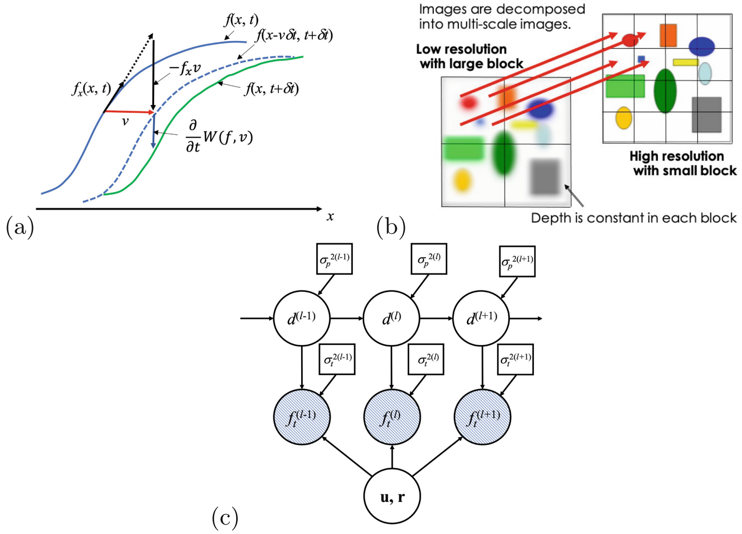


Fig. 2. Features of the proposed method; (a) Compensation calculation of the time derivative of the image brightness based on the warp of the first frame image. For simplicity, only the x coordinate value is shown. The blue dashed line represents the warped image. (b) Multi-resolution and local optimization strategy. (c) Bayesian network representation of our probabilistic model. (Color figure online)

where, $\hat{\boldsymbol{v}} = [\hat{v_x}, \hat{v_y}]^\top$ is an optical flow estimate calculated from the depth and camera motion estimated at this point. The time partial derivative $\partial/\partial t$ in this equation is practically executed as a finite difference, and image warping $\mathcal{W}$ is defined as

$$\mathcal{W}(f, \boldsymbol{v})(\boldsymbol{x}, t + \delta t) \equiv f(\boldsymbol{x} - \boldsymbol{v}\delta t, t + \delta t). \tag{8}$$

The mechanism of this f_t calculation is shown in Fig. 2(a), and the f_t allows for higher resolution depth updates while preventing aliasing in the upper layers. Note that in this study f_x and f_y are calculated for the first frame.

3.4 Hierarchical Depth Model for Local Optimization

There are two main ways to reduce the depth degrees of freedom while avoiding the aperture problem. One is the global optimization method and the other is the local optimization method. The former requires that the optical flow and depth are spatiotemporally smooth. The latter assumes that the optical flow and depth are constant in spatiotemporally localized regions.

For integration with multi-resolution schemes, local optimization methods that reduce the size of the local region as resolution increases are very compatible. No additional processing is required to estimate the location of the depth discontinuity, and the estimated depth automatically retains the depth discontinuity. Defining depth variables in multiple layers increases the number of variables, but defining transition probabilities between layers reduces the effective depth degrees of freedom. This results in a stable depth recovery. Figure 2(b) shows the outline of the processing strategy.

By using this hierarchical depth model, we can calculate the prior probability for the higher resolution layer based on the depth estimation results for the lower resolution layer during inference. This avoids the arbitrariness of the prior probability for each layer. For the lowest resolution layer, we can specify an arbitrary prior probability, in which case it is appropriate to express it in terms of parameters and estimate these parameters as a type-2 maximum likelihood estimator (MLE) [8]. In this study we do not use this prior probability, so overall we do not use any external knowledge of depth.

4 Method

4.1 Definition of Probabilistic Model and Parameters

Depth Model. Layers with different resolutions are identified by the indexes $l = 1, 2, \cdots, L$, where $l = 1$ has the lowest resolution and $l = L$ has the highest one. $d^{(l)}$ represents the shallowness of each pixel with a layer l, and $\{d^{(l)}\}$ is a set of shallowness of all the pixels. The resolution of the depth map recovered at each layer, that is, the size of the region where the depth is constant, should be determined according to the image resolution. The number of the local regions for the layer l is indicated by $R^{(l)}$, and $M_r^{(l)}$ represents the number of pixels in the region r. It is straightforward to keep $M_r^{(l)}$ constant regardless of r. $R^{(l)}$ is set so that $R^{(l_1)} < R^{(l_2)}$ is satisfied for $l_1 < l_2$.

We define a probabilistic model of shallowness. As the depth relationship between layers, we adopt the following model with linear interpolation operator $U^{(l)}$ and perturbation $\epsilon_p^{(l)}$.

$$d^{(l+1)} = U^{(l)} d^{(l)} + \epsilon_p^{(l+1)}. \tag{9}$$

It is assumed that the linear interpolation is calculated using the depth of adjacent regions. If $\epsilon_p^{(l)}$ follows a Gaussian distribution with mean zero and variance $\sigma_p^{2\,(l)}$ which is common to all regions but is independent for each layer, then $d^{(l+1)}$ is a random variable with the following Gaussian distribution.

$$p_d(d^{(l+1)} | d^{(l)}, \sigma_p^{2\,(l+1)}) = \frac{1}{\sqrt{2\pi\sigma_p^{2\,(l+1)}}} \exp\left\{ -\frac{(d^{(l+1)} - U^{(l)} d^{(l)})^2}{2\sigma_p^{2\,(l+1)}} \right\}. \tag{10}$$

This modeling increases the number of unknown variables, but the correlations among them reduce the intrinsic degrees of freedom.

Observation Model. Next, let us consider a probabilistic model of observation. From Eq. 5, the observed $f_t^{(l)}$ of each layer l is modeled by

$$f_t^{(l)} = -\boldsymbol{f}_r^{(l)\top}\boldsymbol{r} - d^{(l)}\boldsymbol{f}_u^{(l)\top}\boldsymbol{u} + \epsilon_t^{(l)}. \tag{11}$$

If $\epsilon_t^{(l)}$ follows a Gaussian distribution with mean zero and variance $\sigma_t^{2(l)}$ which is common to all regions but is independent for each layer, then $f_t^{(l)}$ is a random variable with the following Gaussian distribution.

$$p_f(f_t^{(l)}|d^{(l)}, \sigma_t^{2(l)}, \boldsymbol{u}, \boldsymbol{r}) = \frac{1}{\sqrt{2\pi\sigma_t^{2(l)}}} \exp\left\{ -\frac{(f_t^{(l)} + \boldsymbol{f}_r^{(l)\top}\boldsymbol{r} + d^{(l)}\boldsymbol{f}_u^{(l)\top}\boldsymbol{u})^2}{2\sigma_t^{2(l)}} \right\}. \tag{12}$$

The above-mentioned dependency between observations $\{\boldsymbol{f}_t^{(l)}\}_{l=1,\cdots,L}$ and parameters $\boldsymbol{m} = [\boldsymbol{u}^\top, \boldsymbol{r}^\top]^\top$ and $\Theta_l = \{\sigma_p^{2(l)}, \sigma_t^{2(l)}\}$ can be shown in Fig. 2(c) by the Bayesian network, which is one of the graphical models. Here $\boldsymbol{f}_t^{(l)}$ is the set of $f_t^{(l)}$ for all pixels.

4.2 Information Propagation by Variational Bayesian Inference

Mean-Field Approximation. To be computed are the posterior distributions of $\{\boldsymbol{d}^{(l)}\}_{l=1,\cdots,L}$ and $\boldsymbol{m}$ in the graphical model shown in Fig. 2(c). Here, $\boldsymbol{d}^{(l)}$ is a set of all $d^{(l)}$ in that layer, that is, all $d^{(l)}$ for each region where the depth is assumed to be constant. The MLE of $\{\Theta_l\}_{l=1,\cdots,L}$ must also be inferred. There are various methods for estimating posterior probabilities based on the graphical model. Belief propagation is well known, and its relationship with inference based on mean-field approximation (MFA) is also discussed. In this paper, the latter is adopted to derive a concrete algorithm.

The fundamental strategy is to predict the prior probabilities of $\boldsymbol{d}^{(l)}$ and $\boldsymbol{m}$ used in the current layer from their posterior probabilities obtained in the lower layer. First, consider the joint probabilities of all random variables in a layer.

$$p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{m}|D^{(l-1)}, \Theta_l) = p_f(\boldsymbol{f}_t^{(l)}|\boldsymbol{d}^{(l)}, \boldsymbol{m}, \Theta_l)p_{dm}(\boldsymbol{d}^{(l)}, \boldsymbol{m}|D^{(l-1)}, \Theta_l), \tag{13}$$

where, $D^{(l-1)} \equiv \{\boldsymbol{f}_t^{(l-1)}, \ldots, \boldsymbol{f}_t^{(1)}\}$. In this problem, by using MFA on the prior distribution p_{dm} in layer l predicted from layer $l-1$, MFA can naturally be applied to the posterior distribution in layer l.

$$p_{dm}(\boldsymbol{d}^{(l)}, \boldsymbol{m}|D^{(l-1)}, \Theta_l) \approx q_d(\boldsymbol{d}^{(l)}|D^{(l-1)}, \Theta_l)q_m(\boldsymbol{m}|D^{(l-1)}, \Theta_l). \tag{14}$$

The prediction distribution of $\boldsymbol{d}^{(l)}$ for observations up to $D^{(l-1)}$ is, as in Eq. 10, expressed as

$$q_d(\boldsymbol{d}^{(l)}|D^{(l-1)}, \Theta_l) = \prod_{r=1}^{R^{(l)}} \frac{\exp\{-(d_r^{(l)} - U^{(l-1)}\bar{d}_r^{(l-1)})^2/2\sigma_{e_r}^{2\ (l)}\}}{\sqrt{2\pi\sigma_{e_r}^{2\ (l)}}}, \qquad (15)$$

$$\sigma_{e_r}^{2\ (l)} = {U^{(l-1)}}^2 \sigma_{d_r}^{2\ (l-1)} + \sigma_p^{2(l)}, \qquad (16)$$

where, ${U^{(l)}}^2$ is also the interpolation operator, the weight coefficients of which correspond to the power of each of the corresponding coefficient of $U^{(l)}$. This definition of the variance indicates an approximated representation in which the covariance terms of $d^{(l)}$ are neglected and only the variance terms are considered. Note that $\sigma_p^{2(l)}$ is included as an unknown parameter to be estimated. Equation 15 means that pixel-by-pixel independence of $\boldsymbol{d}^{(l)}$'s prior, that is, $q_d(\boldsymbol{d}^{(l)}|D^{(l-1)}, \Theta_l) = \prod_r^{R^{(l)}} q_{d_r}(d_r^{(l)}|D^{(l-1)}, \Theta_l)$.

On the other hand, since the camera motion is a variable common to all layers, the prior probability can be defined by using the estimation results of the lower layers as they are. For simplicity, we assume that each parameter follows an independent Gaussian distribution. The mean and variance-covariance matrix of these are denoted as $\bar{\boldsymbol{m}}$ and $\boldsymbol{V}_m$, respectively.

$$q_m(\boldsymbol{m}|D^{(l-1)}, \Theta_l) = \frac{\exp\left\{-1/2(\boldsymbol{m} - \bar{\boldsymbol{m}})^\top \boldsymbol{V}_m^{-1}(\boldsymbol{m} - \bar{\boldsymbol{m}})\right\}}{\sqrt{(2\pi)^5 \det\boldsymbol{V}_m}}. \qquad (17)$$

Variational Bayesian Inference. First, the likelihood of the observations appearing in Eq. 13 is given by the following equation.

$$\begin{aligned} &p_f(\boldsymbol{f}_i^{(l)}|\boldsymbol{d}^{(l)}, \boldsymbol{m}, \Theta_l) \\ &= \prod_{r=1}^{R^{(l)}} \prod_{i=1}^{M_r^{(l)}} \frac{\exp\left\{-\left(f_{t(r,i)}^{(l)} + {\boldsymbol{f}_{r(r,i)}^{(l)}}^\top \boldsymbol{r} + d_r^{(l)} {\boldsymbol{f}_{u(r,i)}^{(l)}}^\top \boldsymbol{u}\right)^2 / 2\sigma_t^{2(l)}\right\}}{\sqrt{2\pi\sigma_t^{2(l)}}}, \qquad (18) \end{aligned}$$

where, $M_r^{(l)}$ represents the number of pixels in the region r in layer l. Furthermore, by using Eqs. 14, 15, and 17, Eq. 13, which are the simultaneous probabilities of random variables, are formulated as follows.

$$
\begin{aligned}
&p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{m} | D^{(l-1)}, \Theta_l) \\
&= p_f(\boldsymbol{f}_t^{(l)} | \boldsymbol{d}^{(l)}, \boldsymbol{m}, \Theta_l) q_d(\boldsymbol{d}^{(l)} | D^{(l-1)}, \Theta_l) q_m(\boldsymbol{m} | D^{(l-1)}, \Theta_l) \\
&= \prod_{r=1}^{R^{(l)}} \prod_{i=1}^{M_r^{(l)}} \frac{\exp\left\{ -\left({f_t}_{(r,i)}^{(l)} + {f^r}_{(r,i)}^{(l)} + {f^u}_{(r,i)}^{(l)} d_r^{(l)} \right)^2 / 2{\sigma_t^2}^{(l)} \right\}}{\sqrt{2\pi {\sigma_t^2}^{(l)}}} \\
&\times \prod_{r=1}^{R^{(l)}} \frac{\exp\left\{ -\left(d_r^{(l)} - U^{(l-1)} \bar{d}_r^{(l-1)} \right)^2 / 2{\sigma_{e_r}^2}^{(l)} \right\}}{\sqrt{2\pi {\sigma_{e_r}^2}^{(l)}}} \\
&\times \frac{\exp\left\{ -(\boldsymbol{m} - \bar{\boldsymbol{m}})^\top \boldsymbol{V}_m^{-1} (\boldsymbol{m} - \bar{\boldsymbol{m}})/2 \right\}}{\sqrt{(2\pi)^5 \det \boldsymbol{V}_m}}.
\end{aligned}
\tag{19}
$$

From the principle of variational free energy minimization, we can deduce the log of $q_{d_r}(d_r^{(l)} | D^{(l)}, \Theta_l)$ that is the posterior probability after observing $\boldsymbol{f}_t^{(l)}$ as follows.

$$
\ln q_{d_r}(d_r^{(l)} | D^{(l)}, \Theta_l) = -E_m \left[p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{m} | D^{(l-1)}, \Theta_l) \right], \tag{20}
$$

where, $E_m[\cdot]$ is the expectation with respect to $q_m(\boldsymbol{m} | D^{(l)}, \Theta_l)$. Similarly, the log of $q_m(\boldsymbol{m} | D^{(l)}, \Theta_l)$ can be derived as follows.

$$
\ln q_m(\boldsymbol{m} | D^{(l)}, \Theta_l) = -E_d \left[p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{m} | D^{(l-1)}, \Theta_l) \right], \tag{21}
$$

where, $E_d[\cdot]$ is the expectation with respect to $q_d(\boldsymbol{d}^{(l)} | D^{(l)}, \Theta_l)$. The expectation calculation in Eq. 20 requires q_m derived from Eq. 21, and vice versa. Therefore, both equations are alternately updated until convergence. Note that in this iterative calculation process, $\boldsymbol{f}_t^{(l)}$ is also updated with the updated depth and camera motion according to Eq. 7.

Although $\{\Theta_{l-1}, \cdots, \Theta_1\}$ are not explicitly involved in $\boldsymbol{f}_t^{(l)}$, the estimates of $\boldsymbol{d}^{(l)}$ and $\boldsymbol{m}$ are dependent on $\{\Theta_{l-1}, \cdots, \Theta_1\}$. This implies that it is possible to update $\{\Theta_{l-1}, \cdots, \Theta_1\}$ for $\boldsymbol{f}_t^{(l)}$ observations as well. However, since it is computationally expensive, the estimation of Θ_l is done only in the corresponding l-layer. We can maximize the following Q-function with respect to Θ_l.

$$
Q(\Theta_l) = -E_{dm} \left[p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{m} | D^{(l-1)}, \Theta_l) \right], \tag{22}
$$

where, $E_{dm}[\cdot]$ is the expectation with $q_d(\boldsymbol{d}^{(l)} | D^{(l)}, \Theta_l)$ and $q_m(\boldsymbol{m} | D^{(l)}, \Theta_l)$.

Thus, the entire estimation procedure at each layer constitutes the EM (Expectation-Maximization) algorithm. In the E-step, the depth and motion parameters are updated by VB, and in the M-step, the two variance parameters are updated. See Sect. A for the specific expressions of these equations.

The posterior probabilities of depth and camera motion at layer l are converted and used as the prior probabilities at the next high resolution layer $l+1$, and the averages of these are used to compute $\boldsymbol{f}_t^{(l+1)}$ in Eq. 7. In each iteration of the VB-EM algorithm, f_t can be updated using Eq. 7, but in this study, due to computational cost, it is updated once when the resolution level changes.

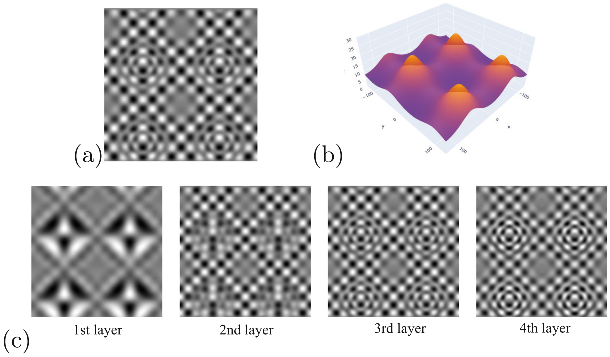


Fig. 3. Data used for experimental evaluation: (a) first image; (b) depth map; (c) results of 4-layer resolution decomposition.

Table 1. Parameter values used for artificial images.

Parameter	Value, etc.
Angle of view	1 × 1 with focal length as 1
Number of pixels	256 × 256
Gradation level	256 greyscale
Number of layers	4
Image decomposition filter DoG filter (Gauss filter for 1st layer)	1st layer: $\sigma = 4$ pix. 2nd layer: $\sigma_1 = 4, \sigma_2 = 8$ 3rd layer: $\sigma_1 = 8, \sigma_2 = 16$ 4th layer: $\sigma_1 = 16, \sigma_2 = 32$
Number of local blocks (block size)	1st layer: 8 × 8 (32 × 32 pix.) 2nd layer: 16 × 16 (16 × 16) 3rd layer: 32 × 32 (8 × 8) 4th layer: 64 × 64 (4 × 4)
Velocity vector	$(u_x, u_y, u_z) = (0.2, -0.2, -0.1)$ $(r_x, r_y, r_z) = (-0.01, 0.0, 0.01)$
Image noise	White additive noise of 3% to the maximum brightness value

5 Performance Verification

5.1 Verification on Artificial Images

The image shown in Fig. 3(a) was generated taking into account the depth shown in Fig. 3(b). The parameters for image generation and resolution decomposition

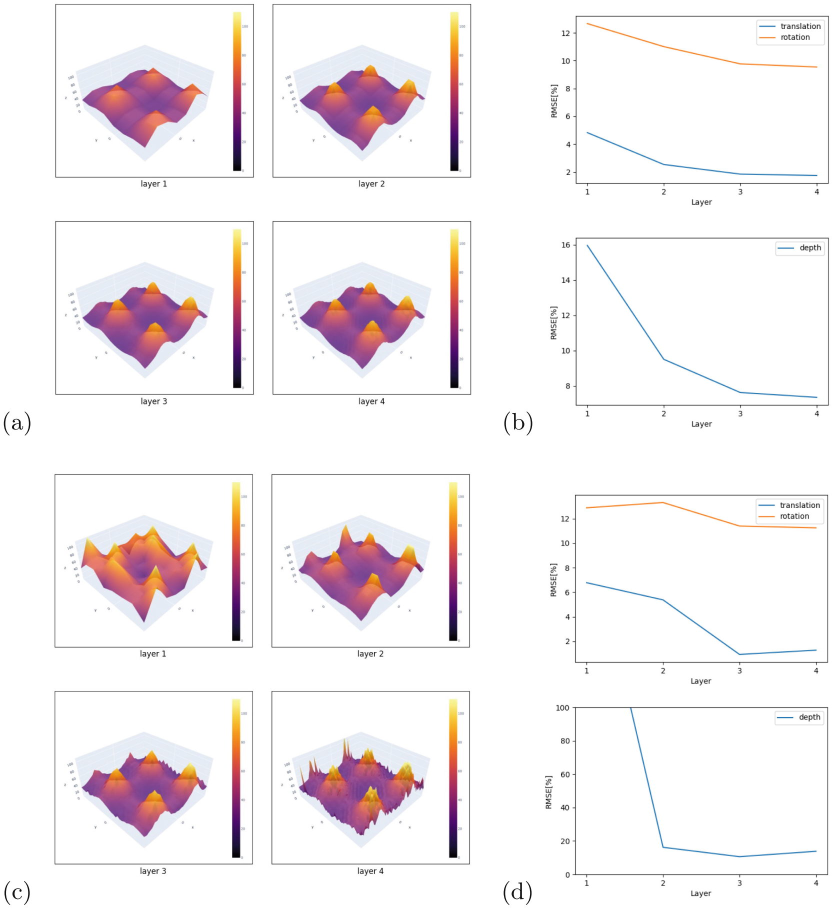


Fig. 4. Hierarchical recovery results for depth from (a) noise-free f_t and (c) f_t computed by Eq. 7. The respective estimation errors are shown in (b) and (d).

are shown in Table 1. A second image was created by changing the texture in accordance with the camera motion and depth settings. Depth is defined as the distance in units of the focal length of the camera. The depth range is from 10 to 40. Under these conditions, the average size of the optical flow is a few pixels. Figure 3(c) shows the result of the resolution decomposition.

In the lowest resolution layer, the prior probabilities of the depth and motion parameters were assumed to be Gaussian with sufficiently large variance to be uninformed prior distributions. This means that the prior probabilities are not used for the whole system.

First, we applied the proposed method to the noise-free f_t, i.e., f_t computed by Eq. 5 at each layer using the true depth and motion parameters. The results confirm the correctness of the proposed algorithm. It is clear from Eqs. 3 and 4

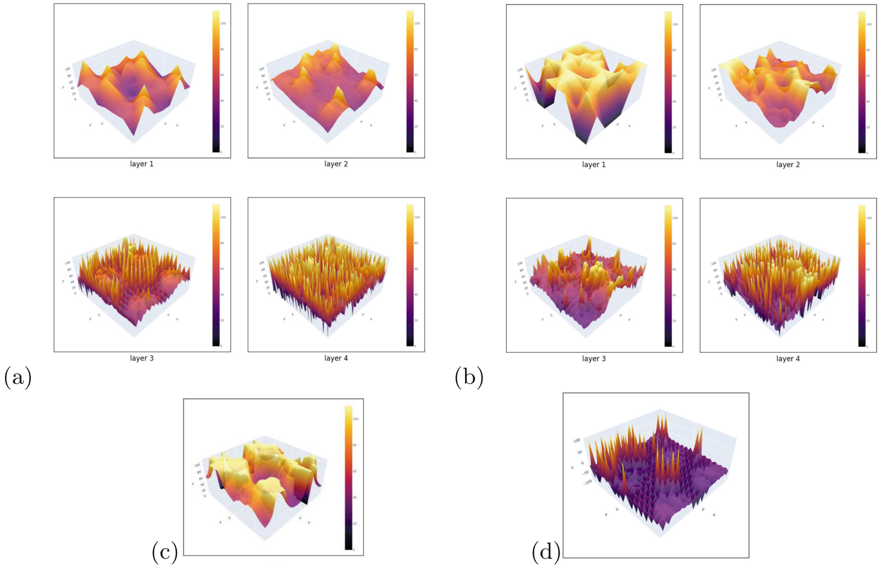


Fig. 5. Ablation studies for the proposed recovery: (a) without image warping; (b) without motion and depth propagation (image warping is done); (c) without motion propagation (EM); (d) MLE without hierarchical processing.

that the norm of $\boldsymbol{u}$ and the scale of d are not uniquely determined. Then we set $\|\boldsymbol{u}\| = 1$.

Figure 4(a) shows that the detailed depth structure is recovered as the resolution increases. In Fig. 4(b), the depth error is defined as the ratio of the root mean square error for pixels to the average depth value. The accuracy of the motion parameters is evaluated by the ratio of the length of the error vector to the length of the true vector. Both errors decrease as processing progresses in the resolution direction. However, the error is not zero despite the absence of noise. This is due to the use of the model for depth. In particular, the fact that the depth is constant in the fourth layer in blocks of 4×4 pixels, which makes it less sensitive to noise, is considered to be a major error factor.

The results from f_t calculated using image warping are shown in Figs. 4(c) and (d). Although the results are clearly worse than Figs. 4(a) and (b), qualitatively looking at the depth recovery results, the proposed method is considered to have sufficient performance. Jagged errors increase from the third to the fourth layer. This means that the error in f_t at the fourth layer is treated as depth information.

Next, we evaluated the recovery performance when part of the procedure in the proposed method is removed. Figure 5(a) shows the case where the image warping is not performed and a simple frame-to-frame difference is used as f_t. A relatively smooth depth is recovered up to the second layer, indicating that aliasing does not occur up to this resolution. However, from the third layer onward, fine irregularities appear, and the recovery error due to aliasing is noticeable, clearly indicating the effect of the warping process. Figure 5(b) shows the results when the image is warped for f_t computation, but both depth and motion parameter estimates are not propagated between the layers. In the third and fourth layers, the degradation of depth estimation is evident, indicating the importance of information propagation by VB. For comparison, Fig. 5(c) shows the results of depth-only interlayer information propagation using the EM algorithm without VB, with the motion parameters updated at each layer in the M-step [14]. This suggests that not only depth but also motion parameters need to be propagated between layers to recover depth detail. The effectiveness of the VB application in this study is obvious. Figure 5(d) is the result of MLE on the 2-frame image without resolution decomposition. Depth was recovered as a constant in 4×4 pixel blocks.

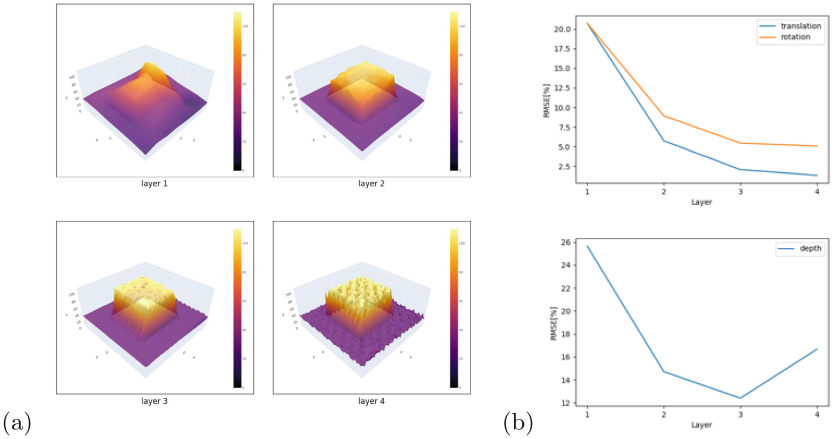


Fig. 6. Recovery of two-planes shape: recovered depth by (a) proposed method and (b) estimation errors for motion and depth.

Table 2. Parameter values changed to real images.

Parameter	Value
Image decomposition filter DoG filter (Gauss filter for 1st layer)	1st layer: $\sigma = 4$ pix. 2nd layer: $\sigma_1 = 4, \sigma_2 = 16$ 3rd layer: $\sigma_1 = 16, \sigma_2 = 64$ 4th layer: $\sigma_1 = 64, \sigma_2 = 128$

Since the depth used in the above evaluations was smooth, the depth of the highly discontinuous shape, consisting of two flat surfaces on the front and back, was also recovered. The images and camera motions used are the same as in the evaluation above. Figure 6(a) shows the recovered depth, and (b) the estimation errors. These results indicate that the proposed method is also effective for edge-preserving recovery.

In this study, the variance of the error in f_t, i.e. σ_t^2, is assumed to be independent for each resolution. In other words, it can handle cases where the noise is not white in the spatial domain. As a result, the number of unknown degrees of freedom increases, and the estimation accuracy of the fourth layer in particular, which has a low signal-to-noise ratio, becomes insufficient. The recovery errors in Fig. 4(c) and Fig. 6(a) are thought to be due to overfitting. By using σ_t^2 as a common parameter for all layers and updating it in the resolution direction using VB in the same way as the motion parameters, it is expected that σ_t^2 can be estimated with high accuracy. This is expected to reduce the depth estimation error, especially in the fourth layer.

It is possible to reapply the VB-EM algorithm to all image information based on the a-priori distributions of depth and motion obtained with the proposed hierarchical method.

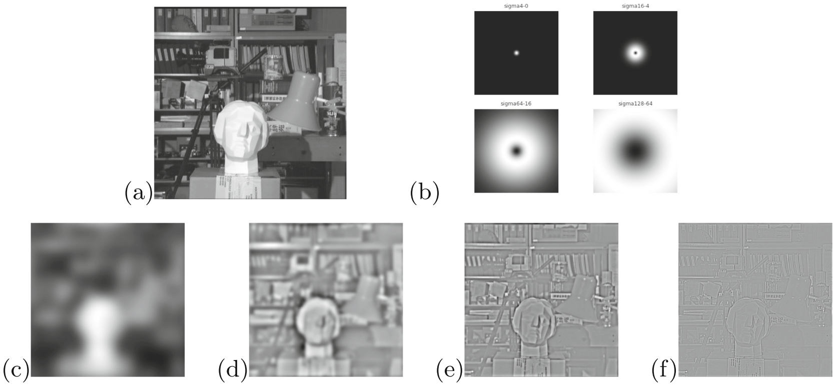
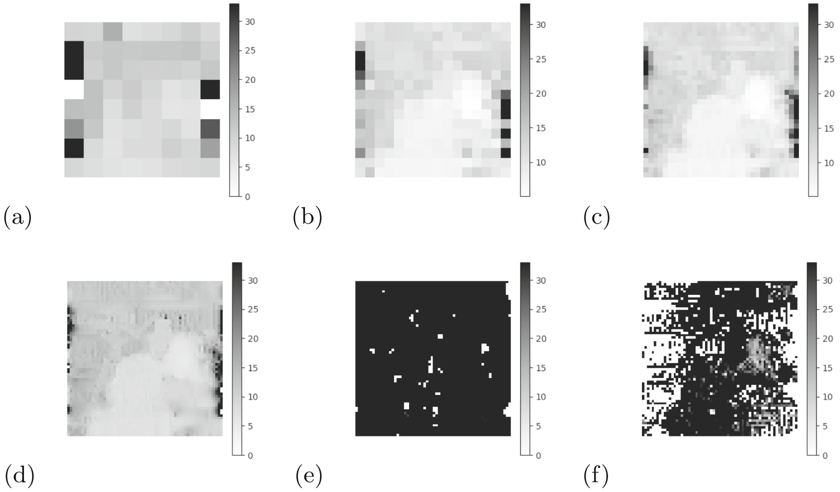


Fig. 7. Actual images used and their resolution decomposition: (a) first frame; (b) frequency characteristics of the resolution decomposition filter; (c)-(f) are the results of resolution decomposition from the lowest resolution to the highest resolution.

5.2 Verification on Real Images

We applied our method to the Tsukuba Stereo Image Dataset - Venus Scene, using the image pair with the smallest baseline. Figure 7 shows the first frame of the image pair, the characteristics of the frequency decomposition filter, and the four decomposed images. We cropped a 256 × 256 pixel area from these images and performed depth recovery on that area. In order to take advantage of the high spatial frequency components that are abundant in this image pair, the DoG filter parameters were set as shown in Table 2.



(a) (b) (c)

(d) (e) (f)

Fig. 8. Results of depth recovery from real images: (a)-(d) are recoveries using the proposed method from lowest to highest resolution; (e) without image warping; (f) MLE without hierarchical processing.

Since this image pair is a standard image for parallel stereo, the translation velocity is only in the x-axis direction, and the rotation velocity is $\mathbf{0}$. We performed depth recovery using the proposed method with these two velocity vectors as unknowns. Figure 8 shows the depth recovery results as a grayscale image. As can be seen from Figs. 8(a) to (d), the proposed method gradually recovers accurate depth from low resolution to high resolution. The results for the fourth layer also include information about the books on the shelf. Figure 8(e) shows the results when no image warping was performed, and Fig. 8(f) shows the results of MLE without hierarchical processing. These are almost entirely black and white, indicating that quantitative recovery was not possible. We also confirmed that the estimation of the motion parameters improves as the information propagates in the resolution direction, and the final values were $\hat{\boldsymbol{u}} = [0.990, 0.125, 0.063]^\top, \hat{\boldsymbol{r}} = [3.99 \times 10^{-3}, -5.93 \times 10^{-3}, -3.91 \times 10^{-3}]^\top$.

In this study, we were able to qualitatively confirm that the proposed method is also effective for real images. However, it is a recovery from small movements between two frames, and the accuracy is insufficient. For accurate depth recognition, it is essential to integrate a large amount of image information. Therefore, it is important to improve the accuracy of two-frame recovery, and the future challenge is to connect the two-frame recovery in this study in the temporal direction. There has been a lot of research on bundle adjustment, and Kalman filter-like online information integration methods have been developed. These are consistent with the method used in this study as a resolution-direction Kalman filter with parameter estimation function, and we will address the extension to the temporal direction as a future issue.

6 Conclusions

In this study, we proposed simultaneous information propagation of depth and camera motion by variational Bayesian inference on a multi-resolution network. The scheme features a hierarchical depth model and image warping based on 3D recovery results in the low-resolution layer, which are properly made to work by variational Bayesian inference. To achieve even higher accuracy, the depth and motion estimation results of the proposed method can be used for image warping, allowing the VB-EM algorithm to be reapplied to all image information in a non-hierarchical manner.

We have only confirmed the effectiveness of the newly introduced techniques and do not refer to the absolute performance of the proposed method. We plan to confirm the effectiveness in detail for real images and then compare it with state-of-the-art methods, especially methods based on robust statistics [19].

The method proposed reduces the intrinsic degrees of freedom of depth, but it still increases with the number of pixels. We have proposed a method to avoid the degradation of camera motion estimation accuracy due to high degrees of freedom in depth [16]. We would like to confirm the effect of incorporating this estimator into the final layer of the method proposed in this study.

Furthermore, we aim to add the "Structure from Shading" function to the proposed method to recover depth and albedo simultaneously. The fusion of methods based on deep neural networks and methods based on the mathematical expression of physical principles, as proposed in this study, is an important issue in early vision problems. In the future, we plan to deepen our investigations, for example, by using deep learning inference as prior information for the proposed method.

A Appendix

Equation 20 is expressed concretely.

$$\ln q_{d_r}(d_r^{(l)}|D^{(l)},\Theta_l) = -\frac{1}{2{\sigma_t^2}^{(l)}}$$

$$\times \sum_{i=1}^{M_r^{(l)}} \Bigg[2 \left\{ \left({f_t}_{(r,i)}^{(l)} + {\boldsymbol{f}_r}_{(r,i)}^{(l)}{}^\top \bar{\boldsymbol{r}} \right) {\boldsymbol{f}_u}_{(r,i)}^{(l)}{}^\top \bar{\boldsymbol{u}} + \langle \boldsymbol{F}_{(r,i)}^{ur(l)}, \boldsymbol{V}_{(r,i)}^{ur(l)} \rangle \right\} d_r^{(l)}$$

$$+ \left\{ \left({\boldsymbol{f}_u}_{(r,i)}^{(l)}{}^\top \bar{\boldsymbol{u}} \right)^2 + \langle \boldsymbol{F}_{(r,i)}^{u(l)}, \boldsymbol{V}_{(r,i)}^{u(l)} \rangle \right\} {d_r^{(l)}}^2 \Bigg] - \frac{\left(d_r^{(l)} - U^{(l-1)} \bar{d}_r^{(l-1)} \right)^2}{2{\sigma_{e_r}^2}^{(l)}} + \text{Const.}, \tag{23}$$

where, $\bar{\boldsymbol{m}} = [\bar{\boldsymbol{u}}, \bar{\boldsymbol{r}}]^\top$ is the mean of q_m, and $\boldsymbol{F}^u = \boldsymbol{f}_u \boldsymbol{f}_u^\top$ and $\boldsymbol{F}^{ur} = \boldsymbol{f}_u \boldsymbol{f}_r^\top$. $\langle \boldsymbol{A}, \boldsymbol{B} \rangle \equiv \mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{B})$ is the Frobenius product (the subscripts (r,i) and (l) are omitted). From Eq. 23, we can see that the posterior probability of $d_r^{(l)}$ has a

Gaussian distribution. The variance is obtained from the coefficient of $d_r^{(l)}$ of the derivative obtained by differentiating the right side with $d_r^{(l)}$.

Equation 21 is represented concretely using $\bar{d}_r^{(l)}$ and ${\sigma_{dr}^2}^{(l)}$ as follows.

$$
\begin{aligned}
&\ln q_m(\boldsymbol{m}|D^{(l)}, \Theta_l) = -\frac{1}{2{\sigma_t^2}^{(l)}} \left\{ \sum_{i=1}^{M_r^{(l)}} \sum_{r=1}^{R^{(l)}} \left({f_t}_{(r,i)}^{(l)} + {\boldsymbol{f}_r}_{(r,i)}^{(l)\top} \boldsymbol{r} \right)^2 \right. \\
&+2 \sum_{r=1}^{R^{(l)}} \bar{d}_r^{(l)} \sum_{i=1}^{M_r^{(l)}} \boldsymbol{u}^\top {\boldsymbol{f}_u}_{(r,i)}^{(l)} \left({f_t}_{(r,i)}^{(l)} + {\boldsymbol{f}_r}_{(r,i)}^{(l)\top} \boldsymbol{r} \right) \\
&\left. + \sum_{r=1}^{R^{(l)}} \left(\bar{d}_r^{(l)2} + {\sigma_{d_r}^2}^{(l)} \right) \sum_{i=1}^{M_r^{(l)}} \left({\boldsymbol{f}_u}_{(r,i)}^{(l)\top} \boldsymbol{u} \right)^2 \right\} - \frac{1}{2} \left(\boldsymbol{m} - \bar{\boldsymbol{m}} \right)^\top \boldsymbol{V}_m^{-1} \left(\boldsymbol{m} - \bar{\boldsymbol{m}} \right) + \text{Const.} \\
&= -\frac{1}{2{\sigma_t^2}^{(l)}} \left(\boldsymbol{m}^\top \boldsymbol{A} \boldsymbol{m} + 2\boldsymbol{b}^\top \boldsymbol{m} \right) - \frac{1}{2} \left(\boldsymbol{m} - \bar{\boldsymbol{m}} \right)^\top \boldsymbol{V}_m^{-1} \left(\boldsymbol{m} - \bar{\boldsymbol{m}} \right) + \text{Const.}, \qquad (24)
\end{aligned}
$$

where, $\bar{\boldsymbol{m}}$ and $\boldsymbol{V}_m$ are the mean and variance of the posterior distribution obtained in one lower resolution layer. $\boldsymbol{V}_m$ indicates the variance-covariance matrix of $\boldsymbol{m}$ with 6 degree of freedom.

Since Eq. 24 is a quadratic equation with respect to $\boldsymbol{m}$, the updated values of $\bar{\boldsymbol{m}}$ and $\boldsymbol{V}_m$ are easily determined. However, $\bar{\boldsymbol{m}}$ must be found by constrained maximization with $\|\boldsymbol{u}\| = 1$, and rank$\boldsymbol{V}_m = 5$ must hold. In this study, $\bar{\boldsymbol{m}}$ was obtained using the Lagrange multiplier method and Newton's method, and $\boldsymbol{V}_m$ was corrected using this $\bar{\boldsymbol{m}}$ so that its rank is 5.

Finally, Θ_l can be updated by maximizing the following Q-function, which is obtained by taking the expectation of $\ln p(\boldsymbol{f}_t^{(l)}, \boldsymbol{d}^{(l)}, \boldsymbol{pd}^{(l)}, \boldsymbol{p}\theta_l)$ in Eq. 19 with q_d and q_m, ignoring the constants.

$$
\begin{aligned}
Q(\Theta_l) = &-\left(\sum_{r=1}^{R^{(l)}} M_r^{(l)} \right) \ln {\sigma_t^2}^{(l)} - \sum_{r=1}^{R^{(l)}} \ln \left({U^{(l-1)}}^2 {\sigma_{d_r}^2}^{(l-1)} + {\sigma_p^2}^{(l)} \right) \\
&- J_{f_t} \left({\sigma_t^2}^{(l)} \right) - J_d \left({\sigma_p^2}^{(l)} \right), \qquad (25)
\end{aligned}
$$

$$
\begin{aligned}
J_{f_t} \left({\sigma_t^2}^{(l)} \right) &= \frac{1}{{\sigma_t^2}^{(l)}} \left\{ \sum_{r=1}^{R^{(l)}} \sum_{i=1}^{M_r^{(l)}} \left({f_t}_{(r,i)}^{(l)} + f^r{}_{(r,i)}(\bar{\boldsymbol{r}}) \right)^2 \right. \\
&\left. +2 \sum_{r=1}^{R^{(l)}} \bar{d}_r^{(l)} f_{(r)}^{ur\,(l)}(\bar{\boldsymbol{m}}) + \sum_{r=1}^{R^{(l)}} \left(\bar{d}_r^{(l)2} + {\sigma_{d_r}^2}^{(l)} \right) f_{(r)}^{u^2\,(l)}(\bar{\boldsymbol{u}}) \right\}, \qquad (26)
\end{aligned}
$$

$$
J_d({\sigma_p^2}^{(l)}) = \sum_{r=1}^{R^{(l)}} \frac{\left(\bar{d}_r^{(l)} - U^{(l-1)} \bar{d}_r^{(l-1)} \right)^2 + {\sigma_{d_r}^2}^{(l)}}{{U^{(l-1)}}^2 {\sigma_{d_r}^2}^{(l-1)} + {\sigma_p^2}^{(l)}}. \qquad (27)
$$

$\sigma_t^{2\,(l)}$ can be computed analytically, while $\sigma_p^{2\,(l)}$ needs to be solved using numerical calculation, for example, the hill climbing method.

References

1. Adiv, D.: Determining three-dimensional motion and structure from optical flow generated by several moving objects. IEEE Trans. Pattern Anal. Mach. Intell. **7**(4), 384–401 (1985)
2. Bae, J., Hwang, K., Im, S.: A study on the generality of neural network structures for monocular depth estimation. IEEE Trans. Pattern Anal. Mach. Intell. **46**(4), 2224–2238 (2024)
3. Bickel, P.J., Klaassen, C.A.J., Ritov, Y., Wellner, J.A.: Efficient and Adaptive Estimation for Semiparametric Models. The Johns Hopkins University Press, Baltimore and London (1993)
4. Farid, H., Simoncelli, E.P.: Optimally rotation-equivariant directional derivative kernels. In: Sommer, G., Daniilidis, K., Pauli, J. (eds.) CAIP 1997. LNCS, vol. 1296, pp. 207–214. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63460-6_119
5. Friston, K.: The free-energy principle: a unified brain theory? Nat. Rev. Neurosci. **11**, 127–138 (2010)
6. Friston, K., Rosch, R., Parr, T., Price, C., Bowman, H.: Deep temporal models and active inference. Neurosci. Biobehav. Rev. **77**, 388–402 (2017)
7. Guizilini, V., Ambrus, R., Chen, D., Zakharov, S., Gaidon, A.: Multi-frame self-supervised depth with transformers. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 160–170. IEEE (2022)
8. Huang, C.T.: Empirical Bayesian light-field stereo matching by robust pseudo random field modeling. IEEE Trans. Pattern Anal. Mach. Intell. **41**(3), 552–565 (2019)
9. Hui, T.W., Chung, R.: Determination shape and motion from monocular camera: a direct approach using normal flows. Pattern Recogn. **48**(2), 422–437 (2015)
10. Ranjan, A., et al.: Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12240–12249. IEEE (2019)
11. Sekkati, H., Mitiche, A.: A variational method for the recovery of dense 3D structure from motion. Robot. Auton. Syst. **55**(7), 597–607 (2007)
12. Sroubek, F., Soukup, J., Zitová, B.: Variational Bayesian image reconstruction with an uncertainty model for measurement localization. In: Proceedings of the European Signal Processing Conference (EUSIPCO). IEEE (2016)
13. Stone, A., Maurer, D., Ayvaci, A., Angelova, A., Jonschkowski, R.: Smurf: self-teaching multi-frame unsupervised raft with full-image warping. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3887–3896. IEEE (2021)
14. Tagawa, N., Kawaguchi, J., Naganuma, S., Okubo, K.: Direct 3-D shape recovery from image sequence based on multi-scale Bayesian network. In: Proceedings of the International Conference on Pattern Recognition (ICPR). IEEE (2008)
15. Tagawa, N., Yang, M.: On computing three-dimensional camera motion from optical flow detected in two consecutive frames. In: Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), pp. 931–942. INSTICC (2023)

16. Tagawa, N., Yang, M.: Epipolar equation weighting for accurate camera motion from two consecutive framesdeep learning for image segmentation. In: de Sousa et al., A.A. (ed.) Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023), vol. 2103, pp. 48–72. Springer (2024)
17. Weng, J., Huang, T.S., Ahuja, N.: Motion and structure from two perspective views: algorithms, error analysis, and error estimation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(5), 451–476 (1989)
18. Xie, Z., Yu, X., Gao, X., Li, K., Shen, S.: Recent advances in conventional and deep learning-based depth completion: a survey. IEEE Trans. Neural Netw. Learn. Syst. **35**(3), 3395–3415 (2024)
19. Yu, F., Zhang, T., Lerman, G.: A subspace-constrained Tyler's estimator and its applications to structure from motion. In: Proceedings of International on Conference Computer Analysis of Image and Patterns (CVPR), pp. 14575–14584. IEEE (2024)
20. Yuille, A., Kersten, D.: Vision as Bayesian inference: analysis by synthesis? Trends Cogn. Sci. **10**(7), 301–308 (2006)
21. Zhu, Y., Cox, M., Lucey, S.: 3D motion reconstruction for real-world camera motion. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE (2011)

3D Pose-Based Evaluation of the Risk of Sarcopenia

Bo-Cheng Liao[1], Jie-Syuan Wu[2](✉), Chen-Lung Tang[1], Gee-Sern Hsu[1], and Jiunn-Horng Kang[2]

[1] National Taiwan University of Science and Technology, Taipei, Taiwan
[2] School of Medicine, College of Medicine, Taipei Medical University, Taipei, Taiwan
b101109092@tmu.edu.tw

Abstract. We propose a computer vision model to assess the risk of sarcopenia in functional movement video clips. Sarcopenia progressively reduces muscle mass and strength with age, posing a significant threat to the well-being of seniors. Early detection and timely intervention can significantly improve an individual's life and alleviate pressure on the healthcare system. Our model includes a 3D posture keypoint detector and a transformer classifier. The posture keypoint detector identifies 16 keypoints that form 4-Vector and 7-Vector input configurations capable of distinguishing individuals with sarcopenia from those without. However, the differences in these configurations between individuals with sarcopenia and those without are too subtle for human observation. Therefore, we trained the transformer classifier to assess the probability of sarcopenia risk in video clips featuring five specific functional movements. We verified our approach through experiments involving 20 sarcopenia patients and 20 individuals without sarcopenia.

Keywords: Sarcopenia · Keypoints · Health · Transformer

1 Introduction

We propose a computer vision model to assess the likelihood of sarcopenia using video clips of human movements. This vision model holds the potential to support healthcare professionals in gauging sarcopenia risk through simple camera-based evaluations. Sarcopenia, a condition commonly associated with aging, involves the gradual and systemic decline in muscle mass and skeletal strength [2]. A study by Duggan et al. [4] underscores the heightened risk of falls among individuals with sarcopenia, with rates as much as double those of their peers without the condition. With up to half of people over 80 years of age affected by sarcopenia [6] and prevalence rates in Asia ranging from 7.3% to 12%, particularly affecting men [8,10,11]. The condition presents a significant health challenge, particularly in aging populations. Addressing sarcopenia

B.-C. Liao and C.-L. Tang—These authors contributed equally to this work. This research is partially funded by NSTC Project No. 111-2221-E-038-006-MY2.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 318–331, 2025.
https://doi.org/10.1007/978-3-031-78444-6_21

through preventive measures and targeted interventions is crucial to mitigate its impact on overall health and quality of life. Our proposed approach distinguishes movement patterns characteristic of sarcopenia patients from healthy individuals and estimates sarcopenia risk based on the captured video footage of subjects' movements.

The current gold standard for diagnosing sarcopenia relies on assessing muscle strength and mass, although these measurements can be challenging to obtain precisely. Muscle strength is typically evaluated using a hand dynamometer [3], while muscle mass assessment involves methods such as dual-energy X-ray absorptiometry (DXA), bioelectrical impedance analysis (BIA), and computed tomography (CT) [3]. However, functional tests also play a crucial role in evaluating body balance and the severity of sarcopenia, aiding in estimating fall risk. We considered five functional tests in our study: Balance A and B, Fukuda, Tandem Gait, and Romberg. We collected a dataset comprising video recordings of 20 patients with sarcopenia and 20 control group individuals performing these tests.

Our approach is based on PoseTriplet [5], a state-of-the-art 3D human pose estimator. PoseTriplet transforms 2D pose sequences into a low-fidelity 3D output, which an imitator then enhances by enforcing physical constraints. These enhanced 3D poses undergo further augmentation by a hallucinator to generate diverse data, which, once again, are processed by the imitator and used to train the pose estimator. This co-evolutionary training scheme enables the development of a robust pose estimator using self-generated data, eliminating dependence on pre-existing 3D datasets.

In the following, we will first present the five functional movement tests and our data set in Sect. 2, followed by our proposed approach presented in Sect. 3, then the experiments in Sect. 4, and then the conclusion of this study in Sect. 5.

2 Functional Movements and Sarcopenia Dataset

The following five functional movements are clinically verified to be good at distinguishing patients with sarcopenia from those without.

- Balance A and B: Functional assessments measure an individual's ability to maintain equilibrium in various positions, engaging key muscle groups such as the abdominal muscles, quadriceps, and thigh adductors. These tests, depicted in Fig. 1, consist of two distinct positions: Balance A and Balance B. Balance A entails assuming a side-by-side stance with feet parallel. At the same time, Balance B requires a semi-tandem stance, positioning the heel of one foot against the side of the other foot's big toe. The objective of these tests is for the individual to sustain an unassisted standing posture for 10 s in each specified stance.
- Tandem Gait (TG): Participants are required to stand with their feet together and carefully walk 10 steps on a flat surface arranged in a straight line, with the specific instruction to walk heel-to-toe. This heel-to-toe walking can be

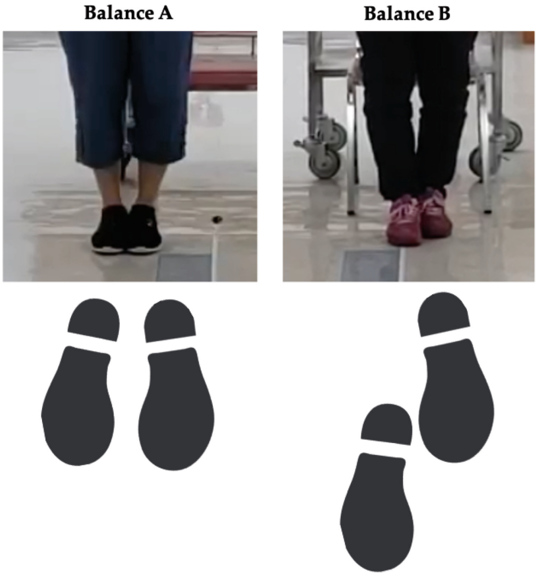


Fig. 1. Balance A, shown on the left, requires standing with feet together side by side. Balance B requires a semi-tandem stand.

subdivided into several movements: abducting the thigh, transferring the center of mass to the other foot, maintaining balance on one foot, adducting the thigh, and transferring the center of mass between the feet. These movements significantly challenge the adductor, abductor, and core muscles. We calculate the steps taken before the first mistake during the test. Figure 2(A) illustrates an example of the Tandem Gait (TG) test. Patients are classified into five grades based on the number of consecutive steps achieved: grade 0 (impossible to walk), grade 1 ($\leq$ 3 steps), grade 2 ($<$ 10 steps), grade 3 (10 steps, but unstable with swaying from side to side) and grade 4 (10 steps, walking normally) [9].

- Fukuda: As Fig. 2(B) shows, the test is a straightforward yet effective tool for assessing balance and mobility. Participants are instructed to stand with closed eyes, arms extended forward, and complete 50 steps in place within designated floor markings. This test engages the quadriceps and biceps femoris muscles for stepping while involving the deltoids and upper arm muscles through arm elevation. The smooth transfer of the center of mass between feet during stepping indicates core muscle strength. By observing the range of motion in the hip joints and limbs, this test provides insights into overall balance capability. Trembling or shaking the limbs often signals inadequate muscle strength, potentially contributing to balance issues.
- Romberg: Fig. 2(C) shows a simple assessment tool for evaluating balance and proprioception. During this test, the patient must stand with their feet together and their eyes closed. This test specifically evaluates the contribution

of proprioceptive and vestibular inputs to balance by eliminating visual input [1]. The examiner closely observes whether the person sways or falls during the test. Any swaying or falling behavior suggests abnormal proprioception, potentially linked to posterior column disease or other neurological conditions affecting balance and spatial awareness.

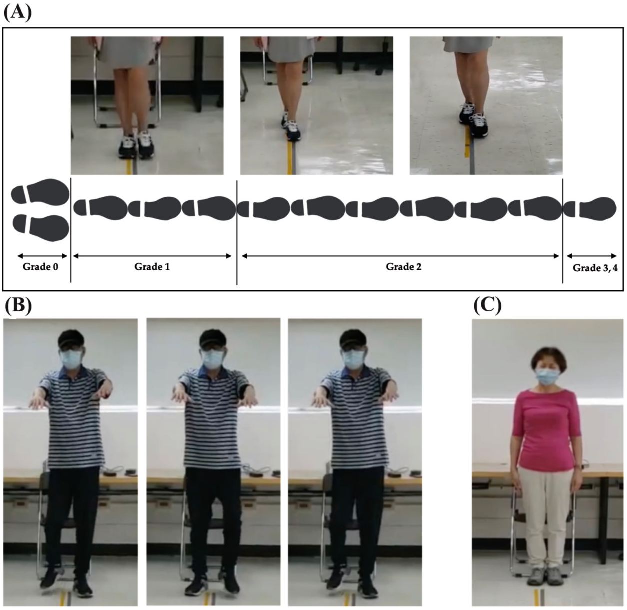


Fig. 2. The figure demonstrates Tandem Gait, Fukuda, and Romberg. As in (A), participants are instructed to walk carefully for 10 steps, with specific guidance to walk heel to toe when performing the Tandem Gait. Based on the number of consecutive steps achieved, we subgrouped patients into five grades: grade 0 (impossible to walk), grade 1 ($\leq$ 3 steps), grade 2 ($<$ 10 steps), grade 3 (10 steps, but unstable with swaying from side to side) and grade 4 (10 steps, walking normally). As shown in (B), Fukuda assesses balance and mobility by having participants close their eyes, lift their hands forward, and take 50 steps within marked floor areas. (C) indicates the Romberg, a straightforward evaluation tool for assessing balance and proprioception. Patients are instructed to stand with their feet together, and their eyes closed.

We recruited 40 individuals, half with sarcopenia and half without, performing five functional tests: Balance A, Balance B, Fukuda, TG, and Romberg. The participants were all older than 20 years.

The studies were conducted according to international ethical guidelines and approved by the Joint Institutional Review Board of Taipei Medical University. All patients provided written consent. The dataset was divided into an 80:20 ratio, with 80% of the subjects allocated for training and the remaining 20% used to test the developed model. Notably, we systematically distributed the dataset across various gender ratios to explore potential gender-related tendencies during the evaluation process and assess the impact of data balance on predictive model performance.

3 Proposed Approach for Estimating Sarcopenia Probability

We formulate the estimation of sarcopenia probability by observing the movements in video clips as a video-based human pose classification. By classifying the video clips of individuals performing the five specified functional movements, both with and without sarcopenia, we develop a pose classifier to differentiate between the two groups. Our approach involves two key components: 3D posture keypoint localization and the classification of these keypoints into distinct categories representing the presence or absence of sarcopenia.

3.1 3D Posture Keypoint Localization

The PoseTriplet [5] is instrumental in pinpointing the 16 keypoints on each subject's body across frames. PoseTriplet consists of a pose estimator, an imitator, and a hallucinator, operating within a dual-loop architecture optimized for joint efficiency. In the initial loop, the pose estimator generates motions that may lack physical coherence, which the imitator promptly corrects by enforcing essential physical constraints, thus producing more physically plausible movements. Subsequently, in the second loop, the hallucinator amplifies the diversity of motions based on the sequence from the first loop, sends them to the imitator for refinement, and further enhances the dataset's richness. This dual-loop paradigm facilitates the coevolution of the three components and enables iterative self-improving training of the pose estimator with the generated diverse yet plausible motion data. Leveraging only 2D pose information as input, the PoseTriplet iteratively refines and hallucinates 3D pose data, nurturing continuous enhancement across all elements within the dual-loop paradigm.

We propose the use of 4- and 7-Vector configurations derived from the 16 3D keypoints detected by PoseTriplet to assess the likelihood of sarcopenia. Clinical observations suggest that the movement and stability of the human trunk during Balance A and B can offer insights into tendencies towards sarcopenia. The 4-Vector configuration, illustrated in Fig. 3, focuses on capturing trunk movements during Balance A/B using four vectors. Similarly, during Fukuda, Tandem Gait, and Romberg, specific body movements signify signs of sarcopenia. In contrast to the four-vector setup for Balance A/B, an additional seven vectors, as depicted on the right side of Fig. 3, constitute the seven-vector configuration to estimate the probability of sarcopenia during Fukuda, Tandem Gait, and Romberg assessments.

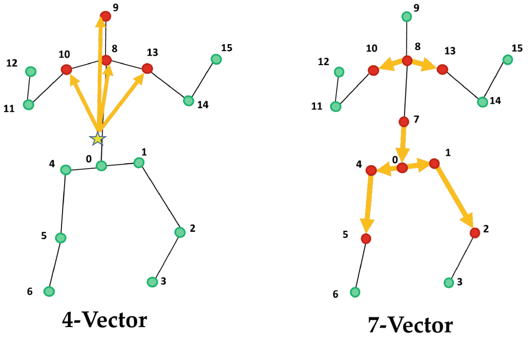


Fig. 3. The feature of 4-Vector system and 7-Vector system. The star point on the left represents the body centroid calculated by keypoints 10, 8, 13, 7, 0, 4, 1, 5, and 2. The orange vector lines represent the chosen features. (Color figure online)

3.2 Transformer Classifier

Figure 4 shows that the key points detected by the PoseTriplet on each image frame constitute the 4-vector and 7-vector inputs. Assuming that we process F image frames in one batch, we denote the 4-Vector input as $V_4 \in R^{F \times 12}$ and the 7-Vector input as $V_7 \in R^{F \times 21}$ for 3D key points. Here, we illustrate using four vectors. Before entering the input embeddings, since the input embeddings consist of three layers of two-dimensional linear transformations, we first convert the original three-dimensional input (Batch, F, Vector) into two dimensions using the x.view() method in PyTorch. This transformation changes the dimensions from three to two, denoted as $V_4' \in R^{F' \times 12}$, where F' is the product of the batch and F. Then, V_4' passes through the three linear layers, becoming f_e', which is represented as $f_e' \in R^{F' \times 128}$. After this, the x.view() method converts the two-dimensional data to three dimensions, represented as $f_e \in R^{F \times 128}$. Before entering the transformer encoder, we enhance f_e with positional encoding from [7] to preserve the ordering information of the sequence at each layer.

f_e then proceeds to the Transformer encoder T_e, which consists of L layers of multi-headed self-attention. Each layer includes a multi-headed self-attention module, which facilitate the capture of global and local attention across the entities of the feature sequence from the previous layer. Let's denote the feature sequences at layer l and $l-1$ as f_l and f_{l-1}, respectively. At layer l, the multi-headed self-attention module initially maps f_{l-1} into a triplet representation comprising the query Q_l, key K_l, and value V_l as follows:

$$Q_l = f_{l-1} W_q, K_l = f_{l-1} W_k, V_l = f_{l-1} W_v \tag{1}$$

Here, W_q, W_k, and W_v are the weights learned during training to determine Q_l, K_l, and V_l, respectively. Next, we compute the dot product of Q_l and K_l using matrix multiplication (matmul), denoted as M_1. After applying the softmax function, denoted as S, to M_1, we obtain N_h attention weights of size $F \times F$. These attention weights are multiplied (matmul) with V_l, denoted as M_2. This

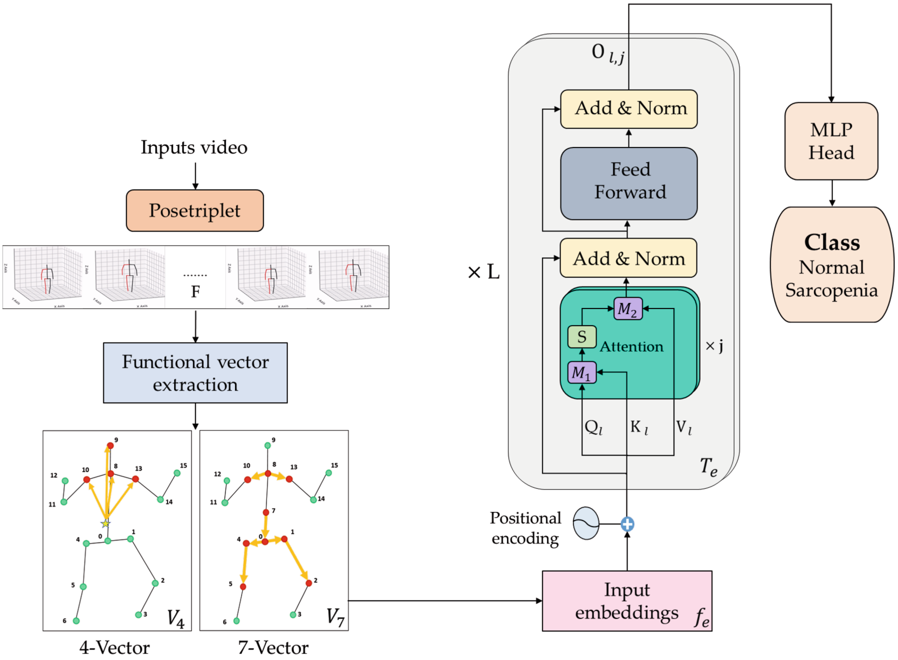


Fig. 4. The SDF network architecture.

process continues for each layer f_l , and ultimately yields an output $O_{l,j}$ from the j-th head is computed as follows:

$$O_{l,j} = \text{softmax}\left(\frac{Q_l^F K_l^F}{\sqrt{d_k/j}}\right) V_l^F, \quad j \in 1, ..., N_h \tag{2}$$

where d_k represents the dimension of K_l, and N_h denotes the number of heads. This learning process repeats for all L layers to yield the self-attention feature sequence f_l. After that, the outputs $O_{l,j}$ from all N_h heads are concatenated and processed by the MLP head layer denoted as $MLP \in R^{F\times128}$ to produce the feature sequence f_m. This MLP layer comprises three linear layers with ReLU activation applied between them. The final step involves a fully connected layer that transforms the prediction probability obtained through the sigmoid function to a range of 0–1. The prediction probability obtained through the sigmoid function is then transformed to a range of 0–1. The standard for diagnosing sarcopenia is based on this probability: a value greater than 0.5 indicates sarcopenia, while less than 0.5 classifies the individual as control group.

4 Experiments

We conducted experiments using the proposed approach on the dataset with 20 patients with sarcopenia and 20 control group. We followed the 5-fold cross-validation and split the data into sets of 80% training (16 subjects per category) and 20% testing (4 subjects per category) in each fold. Evaluation metrics include accuracy, precision, and recall. Accuracy represents the probability of correctly predicting either sarcopenia or control group, precision indicates the accuracy of predicting normal conditions, and recall reflects the accuracy of predicting sarcopenia. Given our emphasis on precisely identifying cases of sarcopenia in evaluating deep neural networks, we primarily utilized accuracy and recall as benchmarks.

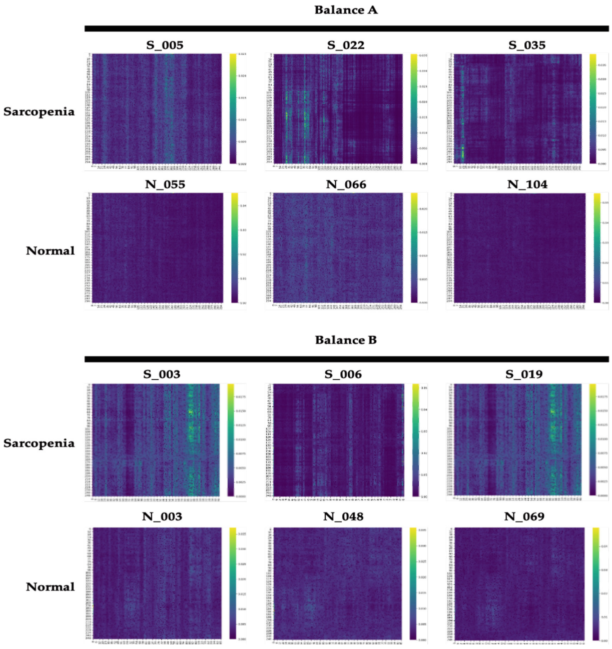


Fig. 5. Samples of attention heatmaps are shown for the 4-Vector configuration, with the upper row representing patients with sarcopenia and the lower row representing those without sarcopenia. The greater color variation across the top heatmaps indicates the distinctions between positive and negative cases. Different variation patterns are shown for Balance A and B.

4.1 Attention Heatmap

Figure 5 displays the attention heatmaps between sarcopenia and control group subjects within the 4-Vector system. For illustration, we selected three examples of sarcopenia patients and three control group from the dataset. The horizontal axis of the attention heatmap corresponds to the timeline, indicating the frame number of the photos.

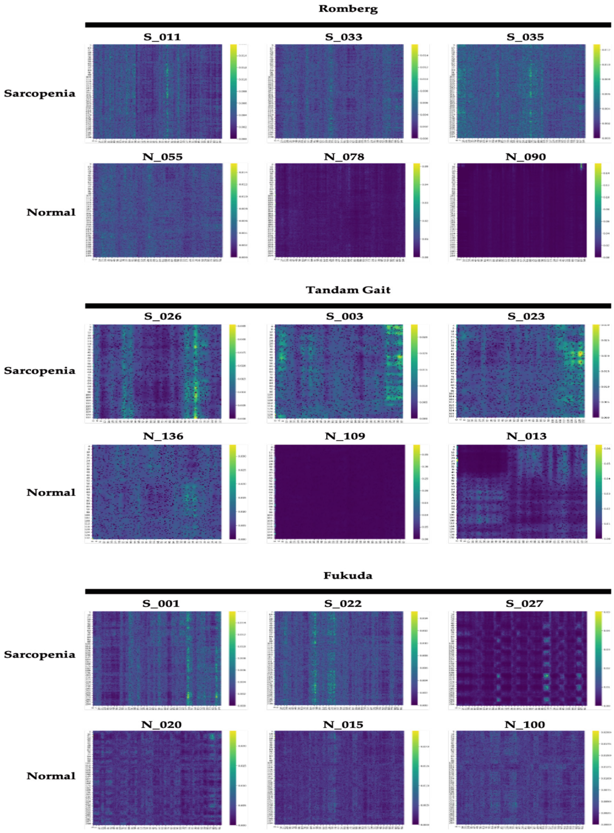


Fig. 6. Samples of attention heatmaps are shown for the 7-Vector configuration, with the upper row representing patients with sarcopenia and the lower row representing those without sarcopenia. The greater color variation across the heatmaps on the top indicates the differences between positive and negative. Different variation patterns are shown for different functional movements.

In the 4-Vector system, the bright spots on the attention heatmap represent the degree of trunk and shoulder sway during the balancing process. In Fig. 5, although the control group exhibits bright green spots indicating body movement, their trunk sway is uniformly distributed across different time points and joints. On the contrary, individuals with sarcopenia show a higher overall number of bright spots on the heatmap, with noticeable periods of significant movement. For example, in patients with sarcopenia during Balance A, individual S_022 exhibits distinct sways in frames 30, 73, and 115, as evidenced by three prominent bright bands on the heatmap.

Figure 6 presents the attention heatmaps within the 7-Vector system. Similar to Fig. 5, the distribution of highlights can be observed, indicating uneven and pronounced trunk movements in individuals with sarcopenia, characterized by more peaks. In contrast, the limb movements of control group appear more uniform. The patterns observed in Figs. 5 and 6 confirm that Our model consists of a 3D posture keypoint detector and a transformer classifier relies on capturing individual trunk sways and enhanced joint movements to assess the likelihood of sarcopenia in patients.

Table 1. Comparative study in 5-fold cross-validation

		Balance A			Balance B			Romberg			Fukuda			Tandem		
	V4	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-
	V7	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓
Accuracy	Fold 1	64%	93%	71%	71%	86%	71%	79%	71%	79%	71%	71%	75%	71%	71%	86%
	Fold 2	57%	86%	64%	71%	93%	71%	71%	71%	79%	86%	83%	86%	64%	75%	79%
	Fold 3	79%	86%	86%	71%	79%	71%	71%	71%	79%	64%	71%	71%	71%	86%	79%
	Fold 4	71%	79%	64%	64%	86%	77%	77%	77%	100%	71%	76%	93%	62%	71%	75%
	Fold 5	68%	76%	71%	68%	83%	75%	71%	71%	86%	73%	73%	83%	64%	71%	79%
	Mean	68%	**84%**	71%	69%	**85%**	73%	75%	72%	**79%**	73%	76%	**87%**	67%	75%	**84%**
Recall	Fold 1	56%	83%	78%	56%	100%	56%	89%	56%	86%	56%	56%	67%	56%	71%	100%
	Fold 2	44%	100%	67%	56%	83%	56%	56%	56%	100%	88%	88%	100%	67%	71%	100%
	Fold 3	75%	71%	75%	56%	86%	56%	56%	56%	70%	67%	56%	75%	71%	86%	88%
	Fold 4	83%	75%	67%	67%	78%	71%	71%	71%	100%	71%	86%	89%	56%	71%	56%
	Fold 5	67%	84%	67%	60%	89%	64%	67%	71%	78%	71%	83%	78%	71%	78%	83%
	Mean	65%	**82%**	72%	59%	**87%**	60%	68%	62%	**87%**	71%	74%	**82%**	64%	76%	**86%**
Precision	Fold 1	83%	100%	78%	100%	100%	100%	80%	100%	75%	100%	100%	75%	67%	71%	67%
	Fold 2	80%	75%	75%	100%	86%	100%	100%	100%	67%	88%	88%	100%	75%	71%	70%
	Fold 3	86%	100%	100%	100%	86%	100%	100%	100%	75%	75%	100%	83%	71%	83%	78%
	Fold 4	63%	86%	75%	75%	100%	83%	83%	83%	83%	71%	75%	75%	75%	71%	75%
	Fold 5	75%	83%	86%	71%	88%	83%	67%	71%	100%	71%	83%	86%	71%	77%	83%
	Mean	77%	88%	83%	89%	92%	93%	86%	91%	80%	81%	89%	84%	72%	75%	79%

4.2 Transformer Classifier

As for the settings of our experiments, we adopted Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the learning rate 2×10^{-5}. All images were resized to 64×128. We trained 12,000 epochs with a batch size of 8, incorporating 32 heads in the multi-head attention mechanism. Our proposed model was implemented in PyTorch and trained on NVIDIA RTX 4090 GPU.

In our experiment, positive samples correspond to cases of sarcopenia. Precision reflects the proportion of predicted cases with sarcopenia, while recall indicates the proportion of actual sarcopenia cases correctly identified by the model. Table 1 shows that in our comparative study, we evaluated different selection methods of 4-Vector and 7-Vector, as well as the scenario without any selection. We used 16 keypoints as input for the non-selection part, representing the entire body skeleton. The results showed that the performance after vector selection was better than using the full-body skeleton. Specifically, 4-Vector performed better for Balance A and B, while 7-Vector showed better results for Romberg, Fukuda, and Tandem gait. Here, we want to understand the impact of

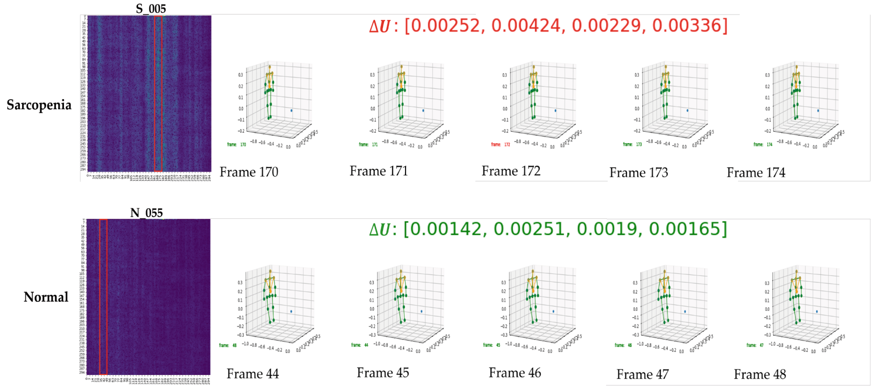


Fig. 7. Compare samples of the 4-vector based on the brightness of frames in the attention heatmap to distinguish between sarcopenia and control group. Bright frames correspond to more considerable Euclidean distances between 3D skeletal vectors in sarcopenia cases (ΔU:[Δu_1,Δu_2,Δu_3,Δu_4]), while dark frames indicate smaller distances in control group individual.

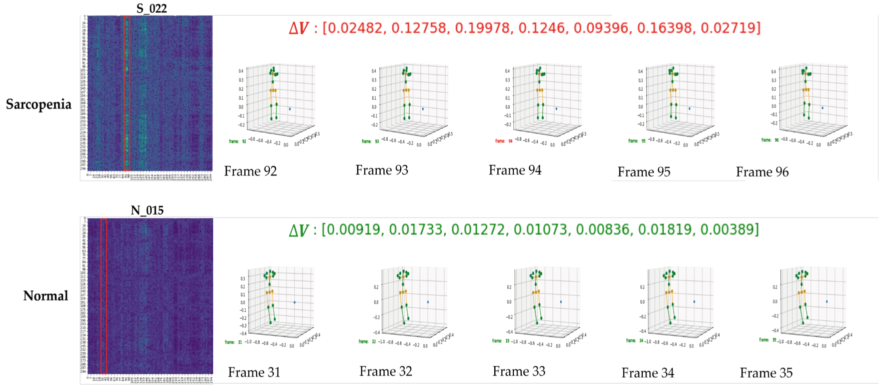


Fig. 8. Compare samples of the 7-vector based on the brightness of frames in the attention heatmap to distinguish between sarcopenia and control group. Bright frames correspond to more considerable Euclidean distances between 3D skeletal vectors in sarcopenia cases (ΔV:[Δv_1, Δv_2, Δv_3, Δv_4, Δv_5, Δv_6, Δv_7]), while dark frames indicate smaller distances in control group.

the attention mechanism in the transformer on vector features. We investigate the segments highlighted in the attention heatmap for 4-Vector and 7-Vector.

Firstly, focusing on the 4-Vector scenario, we illustrate with the "balance A" example. In Fig. 7, we compare cases of sarcopenia and control group. Specifically, for sarcopenia in the "Balance a" scenario, we highlight the example S_005. We extract the brighter frames in the attention heatmap and backtrack to the values of the vector features. By calculating the Euclidean distance between

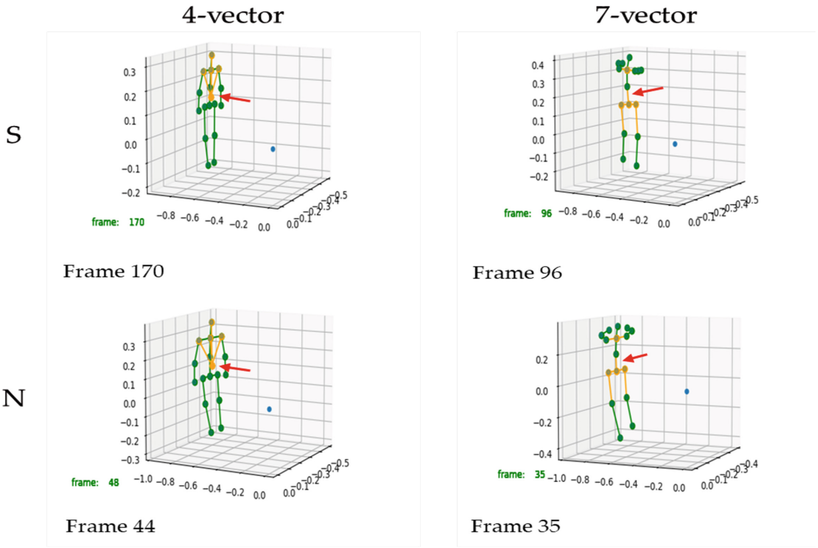


Fig. 9. Samples of 3D vectors were compared between sarcopenia (S) and control group (C) conditions using frames selected from every five frames in Figs. 7 and 8.

each pair of frames, we denote the Euclidean distances for the 4-Vector as ΔU: $[\Delta u_1, \Delta u_2, \Delta u_3, \Delta u_4]$, and for the 7-vector as ΔV: $[\Delta v_1, \Delta v_2, \Delta v_3, \Delta v_4, \Delta v_5, \Delta v_6, \Delta v_7]$. Let's take Δu_1 as an example. The Euclidean distance formula is as follows:

$$\Delta u_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{3}$$

Here, Δu_1 represents the difference between frames for the first vector among the 4-Vector. (x_2, y_2, z_2) denotes the subsequent frame of the first vector, and (x_1, y_1, z_1) denotes the previous frame of the first vector. Then, within a sequence of 5 frames, we calculate the Euclidean distance between each pair of frames and identify the maximum distance as Δu_1. This process is repeated to calculate the differences between frames for the second vector, resulting in Δu_2, and so forth for the remaining vectors.

We observe that frames highlighted in the attention heatmap exhibit larger distances. Conversely, in the case of control group like N_055, frames with darker attention heatmap values show smaller Euclidean distances.

Next, for the 7-Vector scenario, we use "Fukuda" as an example. Figure 8 highlights the example S_022 for "Fukuda". Like the 4-Vector case, frames with brighter attention heatmap values demonstrate more considerable Euclidean distances, represented as ΔV:$[\Delta v_1,\Delta v_2,\Delta v_3,\Delta v_4,\Delta v_5,\Delta v_6,\Delta v_7]$. Conversely, in the case of control group like N_015, frames with darker attention heatmap values exhibit smaller Euclidean distances. This suggests that the attention mechanism can identify frames with more significant movement in the vectors highlighted in the attention heatmap.

Figure 9 shows that we selected one frame from every 5 in Figs. 7 and 8 to compare their 3D vectors. Individuals with sarcopenia show a lower average center of gravity than the control group in the 4-Vector comparison, resulting in a slight forward inclination of the body, leading to more considerable variations ΔU in the 4-Vector compared to those in the control group. In the case of the 7-Vector comparison, the vector from keypoints 0 to 7 (v_1) exhibits a tilting phenomenon in individuals with sarcopenia. At the same time, minimal movement is observed in the same vectors for the control group. Consequently, the values of ΔV are more significant in individuals with sarcopenia than in the control group, corresponding to the brighter regions highlighted in the earlier attention heatmap.

5 Conclusion

In this study, we focus on early detection and intervention of sarcopenia, a condition characterized by age-related decline in skeletal muscle mass, strength, and physical performance. Our approach combines a 3D posture keypoint detector with a transformer classifier, providing a promising strategy for diagnosing sarcopenia from functional test videos.

To enhance interpretability, we integrate mechanisms that offer insights into decision-making. Our approach effectively identifies sarcopenia patients by analyzing body sway and joint movements through attention heatmaps.

Furthermore, our comparative analysis demonstrates the transformer's superior performance in handling the complexity of functional test videos. Our approach achieves higher classification accuracy and precision than traditional approaches, highlighting its potential for more accurate and reliable sarcopenia diagnosis and proactive healthcare interventions.

References

1. Briggs, R.C., Gossman, M.R., Birch, R., Drews, J.E., Shaddeau, S.A.: Balance performance among noninstitutionalized elderly women. Phys. Ther. **69**(9), 748–756 (1989)
2. Cruz-Jentoft, A.J., et al.: Sarcopenia: revised European consensus on definition and diagnosis. Age Ageing **48**(1), 16–31 (2019)
3. Cruz-Jentoft, A.J., Sayer, A.A.: Sarcopenia. Lancet **393**(10191), 2636–2646 (2019). https://doi.org/10.1016/S0140-6736(19)31138-9, https://www.sciencedirect.com/science/article/pii/S0140673619311389
4. Duggan, E., et al.: 121 investigating the relationship between probable sarcopenia and orthostatic hypotension: findings from the Irish longitudinal study on ageing. Age Ageing **51**(Supplement_3), afac218–100 (2022)
5. Gong, K., et al.: Posetriplet: co-evolving 3D human pose estimation, imitation, and hallucination under self-supervision. In: CVPR (2022)
6. von Haehling, S., Morley, J.E., Anker, S.D.: An overview of sarcopenia: facts and numbers on prevalence and clinical impact. J. Cachexia. Sarcopenia Muscle **1**, 129–133 (2010)

7. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
8. Woo, J., Leung, J., Morley, J.: Defining sarcopenia in terms of incident adverse outcomes. J. Am. Med. Dir. Assoc. **16**(3), 247–252 (2015)
9. Yoo, D., Kang, K.C., Lee, J.H., Lee, K.Y., Hwang, I.U.: Diagnostic usefulness of 10-step tandem gait test for the patient with degenerative cervical myelopathy. Sci. Rep. **11**(1), 17212 (2021)
10. Yoshimura, N., et al.: Is osteoporosis a predictor for future sarcopenia or vice versa? Four-year observations between the second and third road study surveys. Osteoporos. Int. **28**, 189–199 (2017)
11. Yu, R., Leung, J., Woo, J.: Incremental predictive value of sarcopenia for incident fracture in an elderly Chinese cohort: results from the osteoporotic fractures in men (mros) study. J. Am. Med. Dir. Assoc. **15**(8), 551–558 (2014)

Learning Explicit Modulation Vectors for Disentangled Transformer Attention-Based RGB-D Visual Tracking

Yifan Pan[1], Tianyang Xu[1](✉), Xue-Feng Zhu[1], Xiaoqing Luo[1], Xiao-Jun Wu[1], and Josef Kittler[2]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, People's Republic of China
{yifan.pan,xuefeng.zhu}@stu.jiangnan.edu.cn,
{tianyang.xu,xqluo,wu_xiaojun}@jiangnan.edu.cn

[2] Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK
j.kittler@surrey.ac.uk

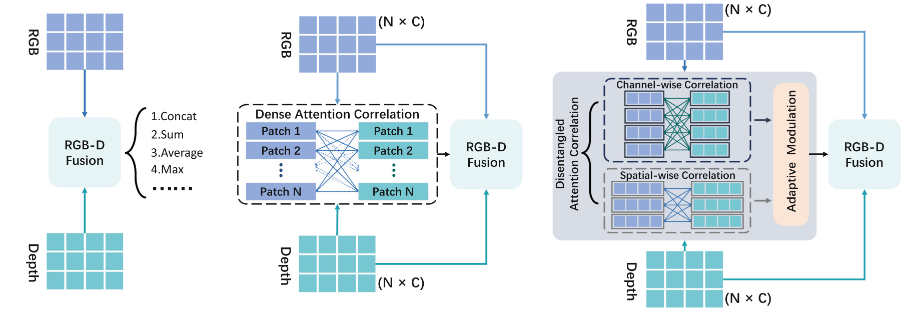
Abstract. Effective fusion of RGB-D multi-modal features is crucial for RGB-D object tracking. Existing fusion methods mainly guide the interaction of RGB and depth by dense attention, but such formulation relies only on independent spatial token attributes, without considering the correspondences among channel slices. To address this limitation, we propose a spatial and channel disentangled attention mechanism, providing dual guidance on spatial and channel relevance for RGB-D fusion. Simultaneously, to deal with potential erroneous attention, we exploit the explicit modulation vectors to weaken less relevant spatial and channel features. Drawing on this, we design an adaptive architecture by weakening the significance of less confident intra-modal features and amplifying the supportive cross-modal features. The experimental results on four standard RGB-D benchmarking datasets, *i.e.*, ARKitTrack, DepthTrack, RGBD1K, and CDTB, confirm the merit of our approach in adaptive fusion, outperforming existing state-of-the-art solutions.

Keywords: RGB-D Object Tracking · Multi-modal Interaction · Disentangled Attention

1 Introduction

Visual object tracking [1,6,13,31,32] is a foundational computer vision task, aiming to predict an object's trajectory within a video sequence based on an initial state. This task holds vital significance for applications like surveillance, autonomous driving, and augmented reality. In general, traditional tracking methods primarily rely on RGB images, utilising colour and texture cues to differentiate targets from the surroundings. However, these frameworks achieve limited performance under challenging conditions, such as low illumination, occlusion, and complex backgrounds, challenging their practical applications.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 332–347, 2025.
https://doi.org/10.1007/978-3-031-78444-6_22



(a) Direct RGB-D Fusion (b) Dense attention-based RGB-D fusion (c) Disentangled attention-based RGB-D fusion

Fig. 1. Motivation of our proposed disentangled attention-based RGB-D fusion approach. Compared to the dense attention-based fusion approach, our approach combines the disentangled attention of the two modalities from the channel and spatial perspectives. Besides, explicit modulation vectors are introduced for both types of attention to weaken the effect of possible noise and redundancy.

To provide a complementary source besides the visual modality, the integration of RGB-D information, obtained from RGB-D sensors capturing both colour (RGB) and depth data, has emerged as a potent solution to these challenges [15,24,26,27,30]. In essence, depth modality offers geometric and spatial alignment of the scene. However, fusing RGB and depth modalities for advanced RGB-D object tracking remains challenging. It is critical to effectively combine these two modalities with boosted and consistent feature representations.

As illustrated in Fig. 1, to explore RGB-D fusion, [24,28] directly fuse the RGB and Depth modalities through feature-weighted averaging or concatenation operations. This direct fusion neglects the inherent differences between the two modalities, potentially compromising the original source information. To this end, recent fusion approaches [9,30] utilise global attention to compute dense attention between the two modalities, thereby integrating the RGB and depth modalities more compactly. However, global attention mechanisms rely more on spatial tokens to establish attention, lacking consideration for channel attributes. Besides, less confident spatial tokens may lead to erroneous attention, resulting in negative information delivery that suppresses the performance of RGB-D trackers. To address this issue, we disentangle the global attention operation into modulatory spatial group attention and modulatory channel group attention. At the same time, we introduce explicit modulation vectors to adjust the relevance of potential errors in the attention computation, adaptively suppressing less relevant spatial and channel features. Based on this, we design an RGB-D interaction module, as shown in Fig. 1(c). Compared to the dense attention-based RGB-D fusion module, our disentangled attention supplements the channel-wise attention between the two modalities. Our disentangled attention-based interaction module ensures effective utilisation of the most informative cues from RGB and depth modalities across different situations, obtaining high-quality fused features that significantly support the localisation demands from RGB-D object tracking.

Overall, we have the following three contributions:

- We disentangle the global dense attention into modulatory spatial and channel group attention to obtain the attention of RGB and depth modal features jointly to guide advanced RGB-D fusion.
- We propose an RGB-D interaction network that combines disentangled self-attention and cross-attention to process both unimodal and bimodal input, providing a new solution for multimodal feature learning.
- We introduce a new RGB-D tracker, DAMT, which updates state-of-the-art performance on four RGB-D object tracking benchmarks.

2 Related Work

2.1 Attention Modules

The introduction of attention modules has significantly improved the power of deep networks to understand image content by selectively focusing on key areas or features. In the beginning, SENet [11] introduces the "Squeeze-and-Excitation" (SE) module, improving the feature discrimination ability of convolutional neural networks (CNN) by dynamically adjusting the importance of channels. Building upon SENet, CBAM [22] further enhances attention to image features by adding spatial attention, enabling the network to identify important channel features and locate crucial spatial features. Simultaneously, the transformer [20], originally designed for natural language processing tasks and effectively adapted to computer vision tasks, computes attention scores between all regions in an image, capturing complex long-range dependencies. This makes it particularly valuable for tasks such as image classification [7] and object detection [3]. Additionally, modules like Dual Attention [8] and Attention in Attention [10] further advance their structures by combining spatial and channel attention mechanisms or applying attention mechanisms recursively at multiple levels respectively.

Different from existing designs, the disentangled transformer attention we propose is mainly aimed at the interaction of RGB and depth modalities in RGB-D object tracking, providing a more delicate way to learn the relevance between RGB and depth modal features.

2.2 Multi-modal Fusion in RGB-D Visual Tracking

Benefited from the rapid advancement of high-performance depth sensors, the emergence of large-scale RGB-D tracking datasets [24,26,27,30] has opened up new possibilities for exploration in RGB-D visual tracking. Compared to traditional RGB object tracking [1,13], RGB-D object tracking integrates information from both RGB and depth images. While this provides more scene information, it also requires a focused consideration of the interaction between the two modalities. For instance, DeT [24], inspired by the architectures of ATOM [5] and DiMP [2], effectively combines RGB and depth features by computing pixel-wise maximum or average values from their feature maps. To exploit transformer structure, SPT [30] initially feeds RGB and depth features into their respective multi-layer transformer encoders, separately processing each modality at the feature extraction stage. Subsequently, it employs channel dimension splicing along with self-attention modules to promote the interaction between RGB and depth features. To preserve the power of the RGB modality-trained tracker, ViPT [29] adopts visual prompt learning, embedding prompt structures into each layer of the interaction module to integrate multi-modal information. To explore the underlying geometric clues, ARKitTrack [26] transforms the depth map into BEV space to obtain a structured depth map coding. Then, Cross-View Fusion is followed to fuse RGB features and depth maps. These RGB-D fusion designs have developed effective innovations in multi-modal feature extraction, feature selection and modal adaptation.

To perform sufficient fusion, our design generates RGB-D multi-modal features from both single-modal feature modulation and multi-modal feature interactive modulation, providing a new and effective solution for RGB-D multimodal fusion.

3 Approach

The framework of our RGB-D tracker DAMT is shown in Fig. 2, where the backbone of the feature extraction and interaction network is the Vision Transformer (ViT) [7]. On this basis, we innovatively develop the Spatial and Channel Disentangled Attention module and create the Disentangled Transformer Attention RGB-D Interaction network. The network excels at integrating high-quality, highly correlated RGB-D features, significantly improving tracking performance. In the following sections, we introduce our DAMT tracker in detail, focusing on three aspects: overall network architecture, Spatial and Channel Disentangled Attention, and Disentangled Transformer Attention RGB-D Interaction network.

3.1 Overall Network Architecture

In this section, we introduce the general architecture of our DAMT tracker. Our bimodal feature extraction and template-search region interaction module is built based on the LiteTrack [21], which is a high-performance tracker of asynchronous feature extraction and interaction. The term "asynchronous" denotes a sequential process where template features are initially extracted, followed by the extraction of the search area features, culminating in the interaction and learning of the template-search region features. Specifically, for the RGB modality, we first input the template image ($\mathbf{Z}_{rgb} \in \mathbb{R}^{C\times H_z \times W_z}$) and the search image ($\mathbf{X}_{rgb} \in \mathbb{R}^{C\times H_x \times W_x}$) into the patch embed layer and convert them into the corresponding token patches ($\mathbf{Z}_{rgb} \in \mathbb{R}^{C\times \frac{H_z}{16} \times \frac{W_z}{16}}$, $\mathbf{X}_{rgb} \in \mathbb{R}^{C\times \frac{H_x}{16} \times \frac{W_x}{16}}$). Feature

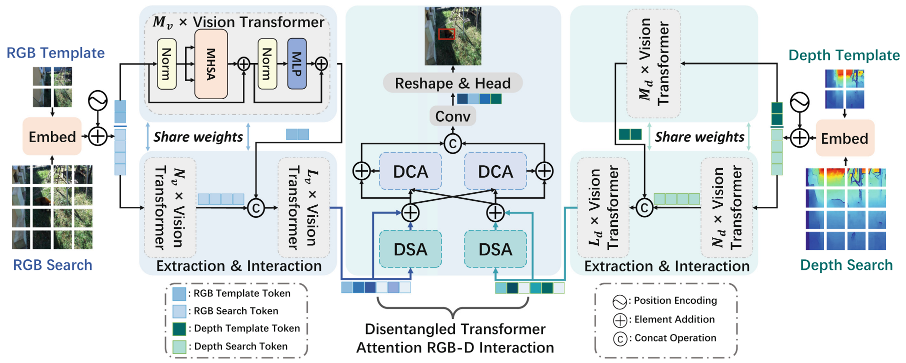


Fig. 2. Overall architecture of our proposed RGB-D tracker DAMT. It includes the RGB Extraction and Interaction branch, Depth Extraction and Interaction branch, and Disentangled Transformer Attention RGB-D Interaction.

extraction and interaction are implemented using continuously stacked transformer encoder layers. The basic block uses ViT [7], and its internal operation is multi-head self-attention:

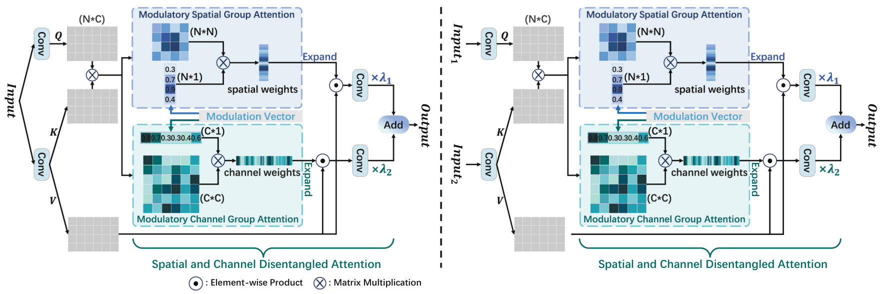
$$\mathrm{Attn} = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{1}$$

For the template patches $\mathbf{Z}_{rgb}$, we send it into M_v ViT blocks for feature extraction. Regarding the search patches $\mathbf{X}_{rgb}$, we first feed it into the corresponding feature extraction phase with N_v ViT blocks to perform the multi-head self-attention operation, and then concat it together with the template feature tokens output from the last layer, forming the input for the interaction phase. This interaction phase comprises L_v ViT blocks and its internal operation differs slightly from the standard multi-head self-attention. Specifically, the generation of **Query (Q)** only relies on search region features, as shown in Eq. (2).

$$\mathrm{Attn}_{[Z:X]} = \mathrm{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V = \mathrm{softmax}\left(\frac{[Q_x]\left[K_x^{\top};K_z^{\top}\right]}{\sqrt{d_k}}\right)[V_x;V_z] \tag{2}$$

Although the attention computation in feature extraction is slightly different from that in the interaction phase, the network parameters are the same. For the same modality, the template branch and the search region branch share the weights. For the depth modality, the computation of the feature extraction and interaction phases is the same as that of the RGB modality, the only difference is that the number of ViT blocks used for the depth modality can be different from the RGB modality, but both satisfy Eq. (3). We performed an extensive ablation analysis of the number of ViT blocks for both modalities, as shown in Table 3.

$$M_{\{v,d\}} = N_{\{v,d\}} + L_{\{v,d\}} \tag{3}$$



(a) Disentangled Self-Attention Module (b) Disentangled Cross-Attention Module

Fig. 3. Design details include Disentangled Self-Attention module, Disentangled Cross-Attention module, and Spatial and Channel Disentangled Attention.

After obtaining the RGB interactive token ($\mathbf{T}_{rgb}$) and the depth interactive token ($\mathbf{T}_{depth}$), input them into the Disentangled Transformer Attention RGB-D Interaction network, which consists of two Disentangled Self-Attention (DSA) modules and two Disentangled Cross-Attention (DCA) modules. The network drives the adaptive interaction of the RGB and depth modalities by Spatial and Channel Disentangled Attention. More detailed framework explanations will be provided in Sect. 3.2 and 3.3.

After passing through the RGB-D Interaction network, we extract the search feature tokens from the fused multi-modal features and input them into the central prediction head. This prediction head comprises three convolution branches: central score classification, offset regression, and width-height regression. The position with the highest confidence in the central score map is chosen as the predicted target position, and subsequently, the bounding box is computed as the final output based on the corresponding regression coordinates. We employ the focal loss [14] for classification. For localisation, we combine the L_1 loss and the generalised GIoU loss [19] as the training objectives. The overall loss function can be expressed as:

$$Loss_{\text{total}} = L_{\text{focal}} + \lambda_{\text{G}} \cdot L_{\text{GIoU}} + \lambda_{\ell} \cdot L_1 \tag{4}$$

where $\lambda_{\text{G}} = 2$ and $\lambda_{\ell} = 5$ are trade-off weights, as suggested in [25], to balance optimisation.

3.2 Spatial and Channel Disentangled Attention

To cope with the lack of consideration of channel attributes in dense attention when fusing RGB and depth modalities, we propose the Spatial and Channel Disentangled Attention, which disentangles the dense attention into an adaptive fusion of modulatory spatial group attention and modulatory channel group attention. Its input dimension is $(N \times C)$, where N stands for the number of patches and C represents the number of channel layers. The structure is depicted in Fig. 3.

In particular, Modulatory Spatial Group Attention, which first calculates the degree of attention of each patch relative to all patches by matrix multiplication of Query (Q) and Key (K), while dividing by $\sqrt{n_k}$ to balance the value of the attention weights, and n_k represents the number of patches involved in the current attention computation. Then, explicit spatial modulation vector($\mathrm{Vec_s}$) of dimension $(N \times 1)$ is introduced, and it is applied to the spatial attention map to contract the group attention and fine-tune the attention for each patch, aiming at weakening the effect of some erroneous relevance in the spatial attention. Next, the weight vector ($\mathrm{Attn_s}$) is expanded to the dimension of $(N \times C)$ and dot-multiplied with the value (V). This method will obtain spatial attention features ($\mathrm{F_s}$) by applying modulatory spatial attention weights to the input features, based on the varying levels of attention received by each patch within the current patch group. The whole process is shown in Eq. (5) and (6).

$$\mathrm{Attn_s} = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{n_k}}\right) \cdot \mathrm{Vec_s} \tag{5}$$

$$\mathrm{F_s} = \mathrm{Attn_s} \odot V \tag{6}$$

The operation of Modulatory Channel Group Attention is similar to that of Modulatory Spatial Group Attention. The degree of attention for each channel relative to all channels is calculated by matrix multiplication of Query (Q) and Key (K), while dividing by $\sqrt{d_k}$ to balance the attention weights, where d_k represents the number of channels involved in the current attention computation. Then, explicit channel modulation vector($\mathrm{Vec_c}$) of dimension $(C \times 1)$ is used to contract the channel group attention and fine-tune the attention for each channel, with the intention of mitigating the influence of noisy weights in the channel attention. The weight vector ($\mathrm{Attn_c}$) is expanded to the $(N \times C)$ dimension and then dot-multiplied with the value (V), thereby producing channel attention features ($\mathrm{F_c}$). The entire process is depicted in Eq. (7) and (8).

$$\mathrm{Attn_c} = \mathrm{softmax}\left(\frac{Q^\top K}{\sqrt{d_k}}\right) \cdot \mathrm{Vec_c} \tag{7}$$

$$\mathrm{F_c} = \mathrm{Attn_c} \odot V \tag{8}$$

The computational complexity of Spatial and Channel Disentangled Attention is $O(n^2 d)$, which is on the same order of magnitude as standard dense attention. Finally, we using the learnable parameters λ_1 and λ_2 to perform non-proportional fusion of spatial attention features ($\mathrm{F_s}$) and channel attention features ($\mathrm{F_c}$) to obtain the final output(F). The specific operation is shown in Eq. (9).

$$\mathrm{F} = \lambda_1 \cdot \mathrm{Conv}(\mathrm{F_s}) + \lambda_2 \cdot \mathrm{Conv}(\mathrm{F_c}) \tag{9}$$

3.3 Disentangled Transformer Attention RGB-D Interaction

Our Disentangled Transformer Attention RGB-D Interaction network consists of two Disentangled Self-Attention (DSA) modules and two Disentangled Cross-

Attention (DCA) modules. DSA and DCA are used to process single-modality input and dual-modalities feature inputs, respectively. For single-modality input, we use DSA for its spatial and channel features for relevance filtering to increase the weight of highly relevant spatial and channel information in the overall feature.

$$\mathrm{T}_{\{rgb,depth\}} = \mathrm{T}_{\{rgb,depth\}} + \mathrm{DSA}\left(\mathrm{T}_{\{rgb,depth\}}\right) \tag{10}$$

For dual-modalities inputs, our method relies on the more detailed relevance of the two modalities in spatial and channel demensions to guide RGB-D fusion, which also smoothes the difference between RGB and depth modalities during fusion to a certain extent.

$$\mathrm{T}_{rgb} = \mathrm{T}_{rgb} + \mathrm{DCA}\left(\mathrm{T}_{rgb}, \mathrm{T}_{depth}\right) \tag{11}$$

$$\mathrm{T}_{depth} = \mathrm{T}_{depth} + \mathrm{DCA}\left(\mathrm{T}_{depth}, \mathrm{T}_{rgb}\right) \tag{12}$$

Finally, the outputs of the two DCA modules are concatenated to obtain the final fused features, the search region tokens are extracted and then sent to the central prediction head to predict the target bounding box.

4 Experiments

Our RGB-D tracker DAMT is implemented in Python 3.8 based on PyTorch 1.13.0 and training and test are performed on a single Nvidia RTX3090 GPU. The tracking speed of DAMT is about 60 FPS on a 3090 GPU.

4.1 Experimental Setting Details

Training. The number of stacks of ViT blocks in the feature extraction and interaction branches of our DAMT tracker is chosen based on the parameter combination that yielded the best results in Table 3, which is $[M_v, N_v, L_v, M_d, N_d, L_d]$ = [9, 6, 3, 6, 3, 3]. Input images are resized to 128 × 128 (template) and 256 × 256 (search area) in the DAMT tracker. Training utilized DepthTrack, RGBD1K, and ARKitTrack datasets (approx. 1,300 RGB-D sequences), with a learning rate of 10^{-4} and a loss function inspired by OsTrack's principles. The training spanned 200 epochs for robustness and accuracy.

Testing. We evaluated DAMT on four challenging RGB-D tracking datasets (ARKitTrack, DepthTrack, RGBD1K, and CDTB), surpassing the state-of-the-art performance achieved by previous trackers.

Evaluation. Our evaluation follows the metrics (Precision, Recall, and F-score) from [16] for long-term tracking assessment. Precision measures overlap between predicted and ground-truth bounding boxes on successful detections, Recall measures the mean overlap ratio between the predicted and ground-truth bounding boxes across frames where the target remains within the camera's view, and the F-score combines both metrics for overall performance.

Table 1. Quantitative Comparison with advanced RGB and RGB-D trackers.

Tracker	ARKitTrack [26]			DepthTrack [24]			RGBD1K [30]			CDTB [15]			Type
	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	
STARK [23]	0.407	0.381	0.393	0.503	0.468	0.485	0.481	0.509	0.495	0.657	0.669	0.663	RGB
OSTrack [25]	0.440	0.440	0.440	0.558	0.582	0.570	0.489	0.516	0.502	**0.713**	0.686	0.699	RGB
MixFormer [4]	0.449	0.421	0.434	0.490	0.454	0.471	-	-	-	0.692	0.664	0.678	RGB
ToMP [17]	0.449	0.433	0.441	0.515	0.495	0.505	-	-	-	0.670	0.683	0.676	RGB
LiteTrack [21]	0.447	0.446	0.447	0.529	0.513	0.521	0.512	0.501	0.507	0.683	0.692	0.687	RGB
DDiMP [12]	0.495	0.413	0.450	0.503	0.469	0.485	0.557	0.534	0.545	0.703	0.689	0.696	RGB-D
ATCAIS [12]	0.389	0.343	0.364	0.500	0.455	0.476	0.511	0.451	0.479	0.709	0.696	0.702	RGB-D
Siam_LTD [12]	-	-	-	0.418	0.342	0.376	0.543	0.318	0.398	0.626	0.489	0.549	RGB-D
TSDM [28]	0.389	0.292	0.334	0.442	0.363	0.398	0.455	0.361	0.403	0.647	0.543	0.591	RGB-D
DAL [18]	0.446	0.329	0.378	0.512	0.369	0.429	0.562	0.407	0.472	0.620	0.560	0.589	RGB-D
DeT [24]	0.428	0.405	0.416	0.560	0.506	0.532	0.438	0.419	0.428	0.674	0.642	0.657	RGB-D
DMT [9]	-	-	-	0.619	0.597	0.608	-	-	-	0.662	0.658	0.660	RGB-D
SPT [30]	0.439	0.439	0.439	0.527	0.549	0.538	0.545	**0.578**	0.561	0.654	**0.726**	0.688	RGB-D
ViPT [29]	0.444	0.447	0.446	0.592	0.596	0.594	0.453	0.472	0.462	0.651	0.721	0.684	RGB-D
ARKitTrack [26]	0.488	0.469	0.478	0.617	0.607	0.612	-	-	-	0.711	0.671	0.691	RGB-D
DAMT(Ours)	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	0.546	**0.565**	0.687	0.719	**0.703**	RGB-D

4.2 Comparison with SOTA RGB-D Trackers

To demonstrate the superiority of our RGB-D tracker DAMT, we perform quantitative and qualitative comparisons with a sufficient number of state-of-the-art RGB-D trackers on four standard RGB-D tracking benchmarks.

Quantitative Comparison. In Table 1 we perform a quantitative comprehensive evaluation of 16 state-of-the-art trackers (including DAMT) on the ARKitTrack [26], RGBD1K [30], DepthTrack [24] and CDTB [15] datasets. Among the competitors, STARK [23], OSTrack [25], MixFormer [4], ToMP [17], and LiteTrack [21] stand out as RGB trackers. The remaining competitors, encompassing both depth and colour branches, include well-known works such as DDiMP [12], ATCAIS [12], Siam_LTD [12], TSDM [28], DAL [18], and state-of-the-art methods like DeT [24], DMT [9], ViPT [29], ARKitTrack [26], and SPT [30]. Comparative analyses show that our DAMT tracker performs superiorly, particularly achieving significant F-scores on the ARKitTrack and DepthTrack datasets.

In the ARKitTrack benchmark test, all three indicators, Pr, Re, and F-score, ranked first. The F-score of DAMT significantly exceeded the second-ranked ARKitTrack tracker by 6.3%. Its F-score is also 12.5%, 10.2% and 9.5% higher than leading short-term trackers like DeT, SPT and ViPT respectively. Furthermore, DAMT demonstrates significant enhancements compared to top long-term RGB-D trackers such as DAL and TSDM, with F-score improvements of 16.3% and 20.7%, respectively.

Performance on the DepthTrack dataset is also commendable, with DAMT leading the way in precision, recall and F-score metrics. It outperforms the top performer, ARKitTrack, with a 1.3%, 2.0%, and 1.7% improvement in precision, recall, and F-score, respectively. In addition, DAMT also shows significant advantages over other advanced trackers such as SPT, DeT and ViPT, with F-score improvements of 9.1%, 9.7% and 3.5%, respectively.

Although the DAMT tracker slightly outperforms other state-of-the-art trackers on the RGBD1K and CDTB datasets, it still achieves the highest F-scores. Comparative results on four large-scale RGB-D benchmarking datasets not only highlight the effectiveness of DAMT, but also show its robustness, providing strong support for its adoption in a variety of tracking scenarios.

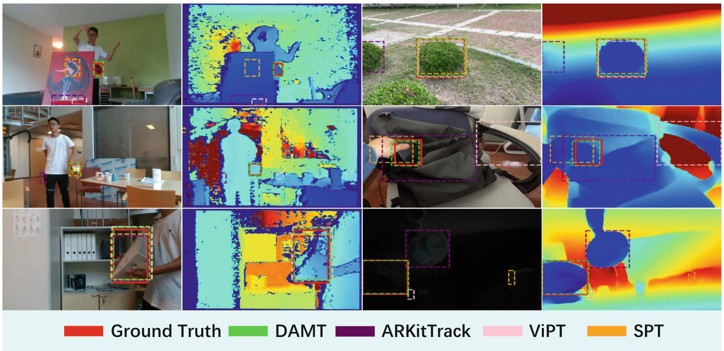


Fig. 4. Qualitative Comparison with state-of-the-art RGB-D trackers.

Table 2. Ablation study on the necessity of RGB-D fusion.

Tracker	ARKitTrack			DepthTrack			RGBD1K			CDTB		
	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score
LiteTrack-baseline	0.447	0.446	0.447	0.529	0.513	0.521	0.512	0.501	0.507	0.683	0.692	0.687
LiteTrack-FT	0.503	0.512	0.508	0.571	0.594	0.583	0.510	0.538	0.524	0.637	0.706	0.670
DAMT	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	**0.546**	**0.565**	**0.687**	**0.719**	**0.703**

Qualitative Comparison. To vividly demonstrate the strengths of our method, we conduct a qualitative analysis comparing DAMT with state-of-the-art trackers, including ViPT [29], SPT [30], and ARKitTrack [26], as shown in Fig. 4. The evaluation is performed on a range of challenging RGB-D sequences, encompassing scenarios such as fast-moving targets, dimly environments, occlusions, and background blending. The visual comparisons vividly highlight the distinct advantages of our proposed method. Our RGB-D tracker remains capable of tracking the corresponding targets in numerous challenging scenarios. This provides compelling evidence of the robustness and effectiveness of our approach in addressing RGB-D object tracking challenges.

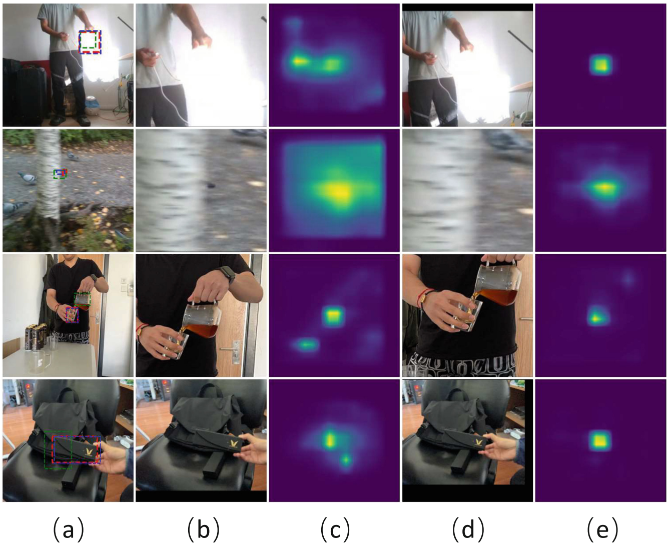


Fig. 5. (a) Target bounding boxes generated by the baseline and DAMT, red represents the ground truth, green is the output of baseline, and blue is the output of DAMT. (b) The search region of baseline. (c) The score map of baseline. (d) The search region of DAMT. (e) The score map of DAMT. (Color figure online)

Table 3. Ablation analysis of ViT Stacks for Feature Extraction and Interaction.

RGB $[M_v, N_v, L_v]$	Depth $[M_d, N_d, L_d]$	ARKitTrack			DepthTrack			RGBD1K			CDTB		
		Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score
[9,6,3]	[9,6,3]	0.525	0.535	0.530	0.619	0.618	0.619	0.575	0.524	0.548	0.681	0.706	0.693
[9,6,3]	[6,3,3]	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	**0.546**	**0.565**	**0.687**	**0.719**	**0.703**
[9,6,3]	[4,2,2]	0.523	0.532	0.528	0.603	0.593	0.598	0.564	0.546	0.554	0.671	0.700	0.685
[6,3,3]	[6,3,3]	0.506	0.504	0.505	0.582	0.556	0.569	0.555	0.512	0.533	0.663	0.690	0.676
[4,2,2]	[4,2,2]	0.488	0.495	0.492	0.526	0.504	0.515	0.523	0.472	0.496	0.658	0.656	0.657

4.3 Ablation Study

Necessity of RGB-D Fusion. Regarding the necessity of RGB-D fusion, in Table 2 we compare three trackers, namely LiteTrack-baseline, LiteTrack-FT and DAMT. LiteTrack [21] is our baseline tracker, and we use the largest public model LiteTrack-B9, LiteTrack-FT is the fine-tuned version of the LiteTrack model, trained on RGB sequences from the DepthTrack [24], RGBD1K [30], and ARKitTrack [26] datasets, and DAMT is our proposed RGB-D tracker. Our method shows improvements in F-score by 9.4%, 10.8%, 5.8%, and 1.6% compared to the baseline tracker, and by 3.3%, 4.6%, 4.1%, and 3.3% compared to the fine-tuned version baseline across four benchmarks. In Fig. 5, we visualize the search region and score map generated by our method and the baseline method for the same scene. As can be seen from the figure, our method exhibits stronger attention towards the tracked target and produces a more accurate

Table 4. Comparing ablation with other attention modules.

Tracker	ARKitTrack			DepthTrack			RGBD1K			CDTB		
	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score
DAMT(SA+CA)	0.528	0.522	0.525	0.599	0.606	0.603	0.558	0.521	0.539	0.675	0.697	0.686
DAMT(DSA+DCA)	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	**0.546**	**0.565**	**0.687**	**0.719**	**0.703**

Table 5. Modular ablation on DSA and DCA in Disentangled Transformer Attention RGB-D Interaction Network.

Dateset	Disentangled RGB-D Interaction			Pr	Re	F-score
	DSA(RGB)	DSA(Depth)	DCA			
ARKitTrack	✓			0.529	0.527	0.528
		✓		0.519	0.528	0.524
	✓	✓		0.532	0.530	0.531
	✓	✓	✓	**0.537**	**0.546**	**0.541**
Depthtrack	✓			0.604	0.607	0.606
		✓		0.591	0.588	0.590
	✓	✓		0.619	0.625	0.622
	✓	✓	✓	**0.630**	**0.627**	**0.629**
RGBD1K	✓			0.553	0.508	0.530
		✓		0.568	0.525	0.545
	✓	✓		0.569	0.533	0.551
	✓	✓	✓	**0.584**	**0.546**	**0.565**
CDTB	✓			0.665	0.693	0.679
		✓		0.671	0.696	0.683
	✓	✓		0.674	0.705	0.689
	✓	✓	✓	**0.687**	**0.719**	**0.703**

Table 6. Ablation Study on methods for Group Attention Contraction.

Attention Contraction	ARKitTrack			DepthTrack			RGBD1K			CDTB		
	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score
Max	0.518	0.528	0.523	0.591	0.582	0.587	0.583	0.539	0.560	0.682	0.701	0.692
Mean	0.519	0.528	0.524	0.592	0.600	0.596	0.577	0.522	0.548	0.680	0.703	0.691
Sum	0.518	0.522	0.520	0.609	0.616	0.612	0.578	0.524	0.550	0.672	0.687	0.679
Modulation Vector	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	**0.546**	**0.565**	**0.687**	**0.719**	**0.703**

Table 7. Ablation study of two weighted adaptive parameters.

Parameters of Add		ARKitTrack			DepthTrack			RGBD1K			CDTB		
Spatial	Channel	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score	Pr	Re	F-score
1	1	0.534	0.532	0.533	0.621	0.625	0.623	0.567	0.528	0.547	0.681	0.711	0.695
λ_1	λ_2	**0.537**	**0.546**	**0.541**	**0.630**	**0.627**	**0.629**	**0.584**	**0.546**	**0.565**	**0.687**	**0.719**	**0.703**

bounding box. These show that effective use of RGB-D bimodal features can improve tracker performance, and our method can well extract and fuse RGB and Depth modalities to achieve significant tracking accuracy and robustness.

Ablation Analysis of the Number of ViT Blocks in Different Modalities. As shown in Table 3, we conducted an experimental analysis on the number of ViT blocks used in the feature extraction and interaction stages for both modalities. Given that the modal information of depth images is relatively simpler compared to RGB images, it might not be necessary to utilise deeper networks for feature extraction.

Comparing Ablation With Dense Attention Modules. As shown in Table 4, We compare our proposed Disentangled Self-Attention and Disentangled Cross-Attention with standard self-attention and cross-attention while ensuring consistency in the training dataset and number of training epochs. In terms of F-score scores, our fusion strategy leads by about 1.6%-2.6%. The results show that the disentangled attention we propose complements the relevance of RGB and depth modalities in channel attributes compared to dense attention, and can provide more detailed guidance for RGB-D fusion.

Modular Ablation on DSA and DCA in RGB-D Interaction Network. We investigate modular ablation of Disentangled Self-Attention and Disentangled Cross-Attention in RGB-D interaction network in Table 5. Our Disentangled Self-Attention optimises the feature quality of a single modality, while Disentangled Cross-Attention can effectively interact with RGB and depth modal features and establish the relevance between RGB and depth modalities from the spatial and channel dimensions. The performance of the DAMT tracker gradually improves as the modules are stacked, which demonstrates the effectiveness and generalisation of our method in fusing RGB-D multi-modal features.

Methods for Group Attention Contraction. To demonstrate that our proposed explicit modulation vectors applied to group attention contraction can better screen out highly correlated spatial and channel features in RGB-D bimodal, we compare the performance of three unified contraction methods(max, mean, and sum). As can be seen from Table 6, our proposed explicit modulation vector contraction can obtain higher F-scores on four benchmarks compared to the other three unified contractions, which indicates the effectiveness of our method.

Ablation of Two Weighted Adaptive Parameters. We perform parameter adaptive summation of features from Modulatory Spatial Group Attention and Modulatory Channel Group Attention. Table 7 provides ablation analysis of learnable weighting (λ_1: λ_2) and equal proportional weighting (1:1). It can be seen from the results that the learnable weighting of spatial and channel features also brings certain performance improvements. The improvement is smaller compared to equal weighting, probably because the use of modulation vectors has weakened most of the irrelevant features.

5 Conclusion

In this paper, we propose a disentangled attention mechanism consisting of modulatory spatial group attention and modulatory channel group attention, which supplements the perceptual deficiencies of dense attention. Based on this, we design a disentangled transformer attention RGB-D interaction network for fusing RGB and depth modalities. Extensive experiments on four RGB-D tracking benchmarks demonstrate that our method exhibits advantages compared to state-of-the-art methods. To further unveil the power of multi-modal fusion, we will explore a fine-grained RGB-D fusion strategy to investigate the RGB-D fusion principles in future.

Acknowledgments. This work was supported in part by the National Key R&D Program of China (2023YFE0116300).
This work was supported in part by the National Natural Science Foundation of China (62106089, 62336004, 62020106012, 62332008).

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14, pp. 850–865. Springer (2016)
2. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6182–6191 (2019)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer (2020)
4. Cui, Y., Jiang, C., Wang, L., Wu, G.M.: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, pp. 18–24 (2022)
5. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: accurate tracking by overlap maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4660–4669 (2019)
6. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: Adaptive context-aware discriminative correlation filters for robust visual object tracking. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 2514–2520. IEEE (2021)
7. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Fu, J., et al.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
9. Gao, S., Yang, J., Li, Z., Zheng, F., Leonardis, A., Song, J.: Learning dual-fused modality-aware representations for RGBD tracking. In: European Conference on Computer Vision, pp. 478–494. Springer (2022)
10. Hao, Y., et al.: Attention in attention: modeling context correlation for efficient video classification. IEEE Trans. Circuits Syst. Video Technol. **32**(10), 7120–7132 (2022)

11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
12. Kristan, M., et al.: The eighth visual object tracking vot2020 challenge results. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, pp. 547–601. Springer (2020)
13. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
14. 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)
15. Lukezic, A., et al.: CDTB: a color and depth visual object tracking dataset and benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10013–10022 (2019)
16. Lukežič, A., Zajc, L.Č., Vojíř, T., Matas, J., Kristan, M.: Now you see me: evaluating performance in long-term visual tracking. arXiv preprint arXiv:1804.07056 (2018)
17. Mayer, C., et al.: Transforming model prediction for tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8731–8740 (2022)
18. Qian, Y., Yan, S., Lukežič, A., Kristan, M., Kämäräinen, J.K., Matas, J.: Dal: a deep depth-aware long-term tracker. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7825–7832. IEEE (2021)
19. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
21. Wei, Q., Zeng, B., Liu, J., He, L., Zeng, G.: Litetrack: layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking. arXiv preprint arXiv:2309.09249 (2023)
22. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
23. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10448–10457 (2021)
24. Yan, S., Yang, J., Käpylä, J., Zheng, F., Leonardis, A., Kämäräinen, J.K.: Depthtrack: unveiling the power of RGBD tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10725–10733 (2021)
25. Ye, B., Chang, H., Ma, B., Shan, S., Chen, X.: Joint feature learning and relation modeling for tracking: a one-stream framework. In: European Conference on Computer Vision, pp. 341–357. Springer (2022)
26. Zhao, H., Chen, J., Wang, L., Lu, H.: Arkittrack: a new diverse dataset for tracking using mobile RGB-D data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5126–5135 (2023)
27. Yang, J., Gao, S., Li, Z., Zheng, F., Leonardis, A.: Resource-efficient RGBD aerial tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13374–13383 (2023)

28. Zhao, P., Liu, Q., Wang, W., Guo, Q.: TSDM: tracking by SiamRPN++ with a depth-refiner and a mask-generator. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 670–676. IEEE (2021)
29. Zhu, J., Lai, S., Chen, X., Wang, D., Lu, H.: Visual prompt multi-modal tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9516–9526 (2023)
30. Zhu, X.F., et al.: RGBD1K: a large-scale dataset and benchmark for RGB-D object tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 3870–3878 (2023)
31. Xu, T., Zhu, X.F., Wu, X.J.: Learning spatio-temporal discriminative model for affine subspace based visual object tracking. Vis. Intell. **1**(1), 4 (2023)
32. Wen, J., Chu, H., Lai, Z., Xu, T., Shen, L.: Enhanced robust spatial feature selection and correlation filter learning for UAV tracking. Neural Netw. **161**, 39–54 (2023)

Attention-Based Patch Matching and Motion-Driven Point Association for Accurate Point Tracking

Han Zang[1], Tianyang Xu[1](✉), Xue-Feng Zhu[1], Xiaoning Song[1], Xiao-Jun Wu[1], and Josef Kittler[2]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, People's Republic of China
{han.zang,xuefeng.zhu}@stu.jiangnan.edu.cn,
{tianyang.xu,x.song,wu_xiaojun}@jiangnan.edu.cn
[2] Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK
j.kittler@surrey.ac.uk

Abstract. Point tracking can be regarded as a transfer and extension of keypoints representation and matching. In contrast to matching the salient points like corners or spots, which are easily detected and described by detector-based approaches, point tracking tasks are capable of handling arbitrary points on physical surfaces, including nonrigid or weakly-textured surfaces. Additionally, keypoint matching lacks a direct mechanism to handle occlusion in tracking tasks. Therefore, we propose to use a detector-free local feature-matching model based on the transformer structure to perform patch matching, incorporating occlusion prediction and introducing an uncertainty estimate for extended robustness. Besides, using a coarse-to-fine strategy, we generate coarse predictions at the patch level and refine them to obtain accurate coordinates at the sub-pixel level by motion-driven point association. After fine-tuning the model on the training set of the Perception Test benchmark, our model APM-MPAPT outperforms the competitors in the benchmark on the corresponding validation set, with promising performance improvement against the baseline.

Keywords: Point Tracking · Feature Matching · Transformer

1 Introduction

Many 3D computer vision tasks, such as Structure from Motion (SfM), and Visual Simultaneous Localisation & Mapping (VSLAM), typically rely on feature matching as a foundational technology. In these tasks, feature matching is primarily employed to associate sparse keypoints between image pairs, as these points often reflect specific physical saliency on rigid objects. Before obtaining the correct correspondences of these keypoints, it is often necessary to detect

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 348–363, 2025.
https://doi.org/10.1007/978-3-031-78444-6_23

and describe them using feature descriptors/maps. However, these detected points in structured and texture-rich areas are expert in 3D reconstruction, rather than depicting the trajectory of moving objects in the scene.

Considering the significance of detecting and predicting motion in scene understanding, point tracking aims to address the problem of long-term motion estimation of points. The targeted points are on arbitrary surfaces of various objects (whether in motion or stationary) within real scenes. Specifically, in this task, the model is given a video along with the 2D coordinates of query points. To this end, there is a basic requirement to address the feature matching problem for query points. Therefore, we attempt to transfer the existing feature matching methods to point tracking by integrating local motion estimation. However, it is worth noting that query points exhibit greater arbitrariness compared to keypoints, which challenges the model to provide discriminative features thereby.

In order to fulfil the demands of tracking arbitrary points, we first perform patch matching between image pairs. These patch features extracted by raw convolutional neural networks (CNNs) tend to focus on local information, neglecting long-range relevance interaction. Therefore, points with repetitive patterns or low texture clues are easily mismatched. To enhance the discrimination of these local features, *i.e.*, to make each patch feature as identical as possible on the same image, and as consistent as possible across image pairs, we employ an attention mechanism [13] to process these patch features for extended global interaction. This mechanism is initially applied to natural language processing tasks and has gradually found its way into computer vision tasks, including image feature matching tasks. For the frame containing the query point, we formulate the objective to match its feature with the corresponding points in subsequent frames of the video. After obtaining a reliable Attention-based Patch Matching (APM) module, we propose to achieve point-level matching. In this stage, motion clues are explicitly studied to construct an interactive shift position prediction module, which performs Motion-driven Point Association (MPA). Finally, we can obtain the model for accurate Point Tracking named APM-MPAPT by combining APM and MPA modules. Moreover, we integrate TAPIR's [2] occlusion prediction of query points and an uncertainty assessment mechanism for extended robustness in our design.

To evaluate our designed APM-MPAPT, we opt for the Perception Test [3] benchmark. As a comparison, we evaluate the performance of the feature matching model LoFTR [1] (same model weights, different matching points of 'interest') on the validation set, and fine-tuned the APM-MPAPT with the off-the-shelf weights, and subsequently evaluate its performance on the validation set. The results obtained from the experiment surpass the static baseline in the benchmark, updating the state-of-the-art record.

The contributions of this paper are three-fold. First, we bridge the gap between existing attention-based feature-matching methods and arbitrary point-tracking techniques, creating a unified approach that leverages the strengths of both methodologies. Second, our method can achieve point tracking across videos of arbitrary length, demonstrating its versatility and scalability. Third, our efficient fine-tuning paradigm requires minimal fine-tuning data and significantly reduces fine-tuning time, ensuring practical applicability for deployment. We have incorporated the corresponding summary into the introduction section.

2 Related Work

Compared to object tracking [28–32], point tracking tasks place a stronger emphasis on feature matching, which is a pivotal step in the process. Considering this, in this section, we introduce related work from the perspectives of point tracking and feature matching.

2.1 Point Tracking

TAP-Vid [20] is a benchmark dataset comprising real-world and synthetic videos with accurately human-labelled and generated point tracks, accompanied by a simple end-to-end point tracking model TAP-Net. In terms of comparing the query point feature with other dense features, TAP-Net generates cross-frame similarity by dot product. The subsequent post-processing via occlusion and coordinate regression branches infer the final position and occlusion degree for TAP-Net. To achieve fine-grained point matching, the Persistent Independent Particles (PIPs) [21] method employs an iterative strategy to gradually infer the correct coordinates and occlusion status. Combining the coarse prediction stage of TAP-Net with the iterative refinement of PIPs, TAPIR [2] designs a two-stage strategy, significantly outperforming all baselines on the TAP-Vid benchmark. This coarse-to-fine pipeline can naturally be generalised to other advanced matching approaches.

To facilitate point tracking evaluation, Perception Test [3] is the benchmarking dataset. This benchmark is designed to assess the transfer capabilities of pre-trained models under zero-shot, few-shot, or fine-tuning conditions. It introduces 11.6k real-world videos, and we only utilise point tracking annotations for evaluation. The benchmark open sources the videos and annotations in the training and validation splits, including 28 training videos with 1,758 tracks, and 73 validation videos with 4,362 tracks. There are two baselines for the benchmark. One is a dummy static baseline which assumes the points never move and are always visible in every frame. With the existence of static points in static camera scenarios, this dummy method also exhibits a certain level of performance. Another baseline uses a TAP-Net model [20] trained on Kubric dataset [25], which directly performs point tracking via cross-frame matching. The benchmark evaluation metrics are as follows. (1) $Position\ Accuracy(< \delta^x)$. (2) $Occlusion\ Accuracy(OA)$. (3) $Jaccard\ at\ \delta$.

2.2 Local Feature Matching

Local feature matching is widely studied in image matching [4]. To establish accurate matching correspondences between image pairs, a considerable amount of work typically adheres to the following stages: detecting keypoints, describing keypoints, matching features of keypoints. To detect points, a straightforward solution is to explore the local saliency (point of interest), ranging from handcrafted designs [5–7] to deep learning approaches [8–10], which only requires identifying sparse correspondences between keypoints that satisfy the geometric reconstruction requirements. Despite the powerful representation of the local salient points, such formulation impedes the requirement of matching any point. Therefore, other methods [1,11,12] that do not rely on feature detection can establish dense feature matching between image pairs, which is suitable to be transferred to point trackers.

However, the above features are extracted by CNNs, suffering from limited receptive fields with poor point identity. Drawing on this, the attention mechanism has been introduced to image feature matching tasks [1,26,27], enhancing the point identity by interacting with long-range features. Furthermore, the positional encoding mechanism enhances the performance of feature matching even in indistinctive regions with weakly textured, motion blur, or repetitive patterns. For instance, COTR [14] is a typical model that employs a transformer architecture to associate image regions by absorbing the most relevant regions with them. It leverages mapping relationships between images to determine the positions of query points on another image.

Both LoFTR [1] and COTR lack a direct mechanism for handing occlusion in point tracking tasks. Additionally, they are trained on MegaDepth [19], a dataset that encompasses real-world scenes and utilises reconstruction to obtain ground-truth correspondence. Hence, they demonstrate more accurate point matching in rigid scenes. To accurately leverage the query point features at various stages of the model, and to enhance occlusion and uncertainty predictions, we opted to transfer and extend the LoFTR model to track points.

To achieve fine-grained point localisation, LoFTR [1] utilises an attention-based [13] architecture for coarse localisation, followed by a fine-grained module to pursue sub-grid precision [2,15–18]. Firstly, it utilises a convolution backbone to obtain the coarse-level and fine-level feature maps. Four stacked self-attention and cross-attention blocks are designed to process the coarse-level feature maps, to obtain rough matches between image patches. Here, the matched patches are determined by the maximal corresponding similarity across the row and column. Subsequently, using these rough matches as anchors, corresponding windows are cropped from the fine-level feature maps. A pair of self-attention and cross-attention blocks is stacked to establish fine-grained heatmap between corresponding windows. Finally, sub-pixel level keypoint correspondences are obtained through the heatmap. However, here, sub-pixel level coordinates are only applicable to one of the image pairs, termed as the ‘queried’ frame, while the other frame, termed as the ‘querying’ frame, still retains integer coordinates. The requirement for integer coordinates has been improved in our model, allowing it to track points at arbitrary sub-pixel positions.

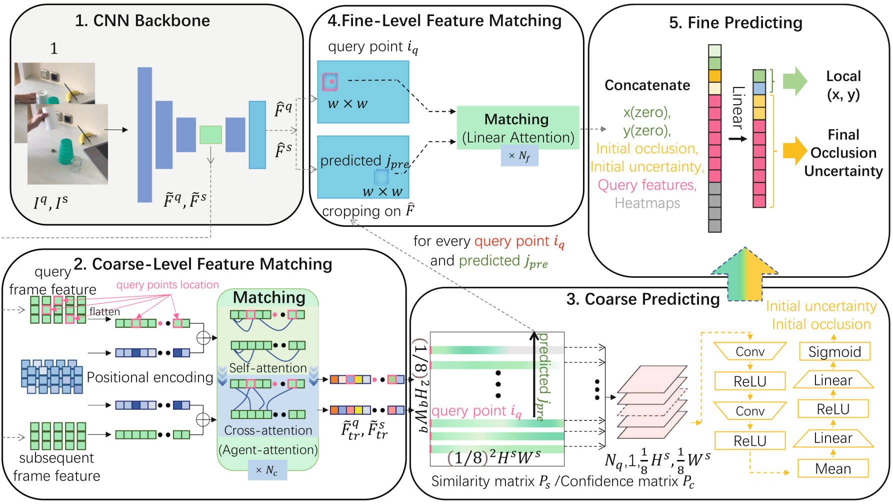
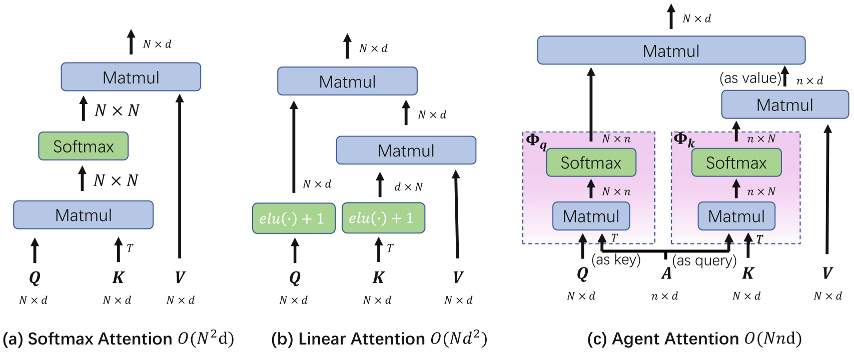


Fig. 1. Overview of our APM-MPAPT. The entire model contains five components: **1.** CNN Backbone extracts the coarse- and fine-level feature maps from two frames. **2.** The coarse feature maps are flattened to 1-D vectors, added with positional encoding, and processed by the Matching module for N_c times. **3.** Using the processed features $\tilde{F}^q_{tr}, \tilde{F}^s_{tr}$ to construct P_s and P_c. From these dense correspondences, locate the rows where the query points resides. The initial uncertainty and occlusion can be preliminarily predicted through neural networks. The max value in every row of P_c is predicted as the most relevant patch, denoted as j_{pre}. **4.** For every query point i_q and its coarse prediction index j_{pre}, a local window with size $w \times w$ is cropped from the fine-level feature maps. **5.** Utilising classification and regression branches, fine predicting module refines all initial predictions to accurately predict the motion.

3 Proposed Approach

The model of point tracking task requires input both a pair of images and the query point coordinates. Leveraging the attention mechanism for dense matching, we precisely ascertain the correlation between the features of the query point and features of another frame. Utilising this information, we devised a coarse-to-fine point tracking method APM-MPAPT. We simultaneously match all the query points, which are annotated within the same frame referred to as the query frames. By conducting matching the query frame with every subsequent frame in the video sequence, we accurately predict the motion of points. Therefore, the model processes only two frames: frame I^q containing the query point and the subsequent frame I^s. An overview of the proposed approach is presented in Fig. 1.



(a) Softmax Attention $O(N^2\mathrm{d})$ (b) Linear Attention $O(Nd^2)$ (c) Agent Attention $O(Nnd)$

Fig. 2. Three computation graphs of different attention mechanisms. (a) Softmax Attention in vanilla Transformer [13]. **(b) Linear Attention** in LoFTR [1] reduces computational complexity. **(c) Agent Attention** [23] uses an agent A pooled from Q to combine high expressiveness with low computational complexity.

3.1 CNN Backbone

In the stage of image feature extraction, we use the standard convolutional architecture with FPN [22] (denoted as CNN Backbone). Through this backbone, pixel values are encoded into higher-dimensional patch features. This process not only captures information at various scales but also ensures a manageable computation cost for subsequent calculations. We denote the coarse-level features of I^q and I^s as $\tilde{F}^q$ and $\tilde{F}^s$, respectively. Similarly, the corresponding fine-level features are denoted as $\hat{F}^q$ and $\hat{F}^s$.

3.2 Coarse-Level Feature Matching

Positional encoding is added to the flattened coarse feature maps and indicates the spatial locations in the Matching module. In this module, we interleave the self and cross attention layers for $N_c = 4$ times with reference to LoFTR [1] to obtain the transformed features, denoted as $\tilde{F}^q_{tr}$, $\tilde{F}^s_{tr}$.

We briefly introduce the attention mechanism in transformer here as background. The attention in transformer plays a key role, as it facilitates the interaction between the query features $f_i(Q)$ and the queried features $f_j(V)$. During the self-attention process, both f_i and f_j originate from the same frame, whereas in the cross-attention, they originate from different frames. Before the interaction, V uses features K as its key features, and typically, K and V are the same. The computation procedure of vanilla self-attention [13], *i.e.*, the Softmax Attention in Fig. 2(a), is as follows:

$$\text{Attention}^S(Q, K, V) = \text{softmax}(QK^T)V \tag{1}$$

Considering the $O(N^2d)$ complexity in the Softmax Attention, LoFTR [1] proposes the more efficient Linear Attention as follows, which utilises the associativity of matrix multiplication to reduce the computational complexity from $O(N^2d)$ to $O(Nd^2)$, since $N \gg d$.

$$\text{Attention}^L(Q,K,V) = \phi(Q)(\phi(K^T)V) \tag{2}$$

where N is the number of all query or value tokens (patches), and d is the hidden dimension of patches. Since the query features Q and the queried features V are derived from the same video, the number of query and value tokens N is equal. Although the model's computational complexity is significantly reduced, there are still limitations in its feature representation capacity. By employing a graceful fusion of Softmax Attention and Linear Attention, Agent Attention [23] effectively combines the advantages of both strong representation capacity and low computational complexity. The computation procedure of the distinct attention is illustrated in Fig. 2(c), and can be written as:

$$\text{Attention}^A(Q,A,K,V) = \text{Attention}^S(Q,A,\text{Attention}^S(A,K,V)) \tag{3}$$

where A, serving as an agent of Q, participating in the computation with a smaller scale, can be obtained by pooling Q. Specifically, for matrix, $n = 49$, indicating that we pool the feature map of Q to a size of 7 × 7.

3.3 Coarse Predicting

Subsequently, a dense similarity matrix P_s can be obtained from the transformed features $\tilde{F}^q_{tr}$, $\tilde{F}^s_{tr}$ with a learnable matrix E, which is initialised as an identity matrix:

$$P_s(i,j) = \frac{1}{\tau} \cdot \langle\langle \tilde{F}^q_{tr}(i), E\rangle, \tilde{F}^s_{tr}(j)\rangle \tag{4}$$

The matrix P_s represents the similarity between features at any position on the query frame and those on the subsequent frame at the coarse level. The N_q rows are located in the similarity matrix based on the coordinates of N_q query points on the query frame. Using this information of N_q rows, we design a coarse prediction module to roughly predict occlusion and uncertainty, drawing inspiration from TAPIR [2]: The Average Jaccard [20] metric is more adversely affected when the algorithm predicts a significantly incorrect location compared to simply marking the point as occluded. Uncertainty u is the other output logit together with occlusion o. It measures the uncertainty of the predicted point coordinates and is trained in a self-supervised manner. We require that the value approaches 1 if the distance between the predicted coordinates and the ground truth exceeds a threshold $\delta = 8$, even if the model predicts that it's visible. We directly employ the coarse-level loss $\mathcal{L}_{cou}(p,o,u)$ in TAPIR to a pair of frames, where $\mathcal{L}_{cou}$ is defined as:

$$\begin{aligned} \mathcal{L}_{cou}(p,o,u) &= \text{BCE}(\hat{u},u) * (1-\hat{o}) + \text{BCE}(\hat{o},o) \\ \text{where,} \qquad \hat{u} &= \begin{cases} 1 & \text{if } d(\hat{p},p) > \delta \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

where $\hat{p} \in \mathbb{R}^2$ and $\hat{o} \in \{0,1\}$ are the ground truth point locations and occlusion on the subsequent frame respectively, when point is occluded $\hat{o} = 1$, d is Chebyshev distance, δ is the distance threshold, and BCE is binary cross entropy.

To enhance the consistency of the feature matching, we then apply the dual-softmax operator [1] on both dimensions of the similarity matrix P_s to calculate the Confidence matrix P_c:

$$P_c(i,j) = \mathrm{softmax}(P_s(i,\cdot))_j \cdot \mathrm{softmax}(P_s(\cdot,j))_i \tag{6}$$

Clearly, P_c and P_s are of the same size. We represent P_c and P_s at the same position in the third part in Fig. 1 for convenience. Then we initialise predictions for the query points coordinates. It's worth noting that when obtaining the N_q row where the query points are located, we retain its fractional part at the current scale, i.e., 1/8 of the original image size. According to P_c and all N_q rows. The maximum value in each row serves as the corresponding coarse prediction j_{pre}:

$$\text{predicted } \; j_{pre} = \arg\max(P_c(i_q,\cdot))_j \tag{7}$$

When mapping back to the coarse-level image coordinates based on the predicted j_{pre}, it is necessary to add the previously retained fractional part. And this added coordinates are the initial coordinates from the module of coarse predicting.

In the coarse prediction, we use the ground truth to obtain matrix $M_c^{gt} \in \{0,1\}$, which has the same size as P_c with $(1/8)^2 H^q W^q \times (1/8)^2 H^s W^s$, and we calculate loss over P_c. The 1 value only appears on some column in the row of the query point i_q and the column must be its correspond visible ground truth j_{gt}. We employ focal loss [24] to compute. We focus only on all value in rows of all query points i_q and all value in columns of all visible ground truth j_{gt}, and the set of $(i_q,\cdot) \cap (\cdot, j_{gt})$ is denoted as F. The loss $\mathcal{L}_{cm}$ calculation is as follows:

$$\mathcal{L}_{cm} = \mathcal{L}_{cp} + \mathcal{L}_{cn} \tag{8}$$

$$\begin{gathered}\mathcal{L}_{cp} = -\frac{1}{||M_c^{gt}||_1} \alpha \sum (1 - P_c(i,j)^{\beta} \log(P_c(i,j)) \\ \text{s.t. } \; (i,j) \in F \;\text{ and }\; M_c^{gt}(i,j) = 1\end{gathered} \tag{9}$$

$$\begin{gathered}\mathcal{L}_{cn} = -\frac{1}{||M_c^{gt}(i,j) - 1||_1} (1-\alpha) \sum (P_c(i,j)^{\beta} \log(1 - P_c(i,j)) \\ \text{s.t. } \; (i,j) \in F \;\text{ and }\; M_c^{gt}(i,j) = 0\end{gathered} \tag{10}$$

The parameters α and β in Eqs. (9) and (10) are two hyperparameters for focal loss, with α typically set to 0.25 and β set to 2.

3.4 Fine-Level Feature Matching

This part primarily deals with fine-level features $\hat{F}^q$ and $\hat{F}^s$. For every query point i_q and its correspond predicted j_{pre}, we crop the size of $w \times w = 7 \times 7$ windows with the center of their coordinates, and then we use bilinear interpolation to obtain more accurate features. We match these N_q pairs of 7×7 windows with $N_f = 1$ time self-attention and cross-attention, which are Linear attention architecture. Simultaneously, we upsample the fine-level features by a factor of two and perform the same operation. We use the transformered features from fine- and its upsampling- level to obtain N_q pairs of 7×7 heatmaps. Each pair of heatmaps are unfolded and concatenated for subsequent processing. For two levels of transformered query features, we merge them into one set of features with the same dimension as the fine-level features.

3.5 Fine Predicting

For every query point, we first concatenate the prepared information as follows: two zero initialisation coordinate (x, y), the initial occlusion and uncertainty from coarse predicting module, the integrated query features (128 dimensions) from fine-level feature matching, and the flatten and concatenate heatmaps from fine-level feature matching. So for every query point, the concatenated feature has $2{+}2{+}128{+}2{\times}7{\times}7$ dimensions. We processed this feature with a linear layer and obtained a feature with $2 + \ 2 + 128$ dimensions. The first two dimensions respectively correspond to the local coordinates x, y. After further processing, adding them to the initial coordinates prepared from coarse predicting module yields the final predicted coordinates (x, y). The features beyond the second dimension are processed in another branch, resulting in the final estimates for occlusion and uncertainty. The loss for the fine-level predicting is given by:

$$\begin{aligned} \mathcal{L}_f(p, o, u) = \ & \mathrm{BCE}(\hat{u}, u) * (1 - \hat{o}) \\ + \ & \mathrm{BCE}(\hat{u}, o) \\ + \ & \mathrm{Euc}(\hat{p}, p) * (1 - \hat{o}) * (1 - \hat{u}) \\ \text{where,} \qquad \hat{u} = & \begin{cases} 1 & \text{if } \ d(\hat{p}, p) > \delta \\ 0 & \quad \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

The meaning of all the variables in Eq. (11) is the same as in equation (5), and the Euc represents the Euclidean metric. It is noteworthy that fine-level predictions are only relevant to the mutual comparison within local regions. Therefore, we compute distance loss only for distances within the threshold δ. Additionally, when the value $\hat{u}$ is 1, i.e., $d(\hat{p}, p) > \delta$, we interpret it as the query point being occluded in local window of another frame. This may be different from the occlusion ground truth, therefore, we directly use $\hat{u}$ as the supervision signal for occlusion loss calculation.

Follow TAPIR [2], in order to comprehensively consider occlusion and uncertainty predictions, we do a same soft combination of two probabilities: the algorithm outputs that the point is visible if $(1 - u) * (1 - o) > 0.5$.

3.6 Objective

The final loss consists of the losses for the coarse- and fine-level:

$$\mathcal{L} = w_{cm}\mathcal{L}_{cm} + w_{cou}\mathcal{L}_{cou} + w_f\mathcal{L}_f \tag{12}$$

On the training set of the Perception Test [3] benchmark, we transform the trajectories of points to correlate with each frame in the video. This way, when selecting any two frames within a video, we can simultaneously obtain the coordinates of any trajectories appearing, along with the ground truth for occlusion. Simultaneously, we map these coordinates to the positions of feature patches at the coarse level, facilitating the construction of M_c^{gt} used to supervise P_c. At the same time, we retain the decimal part of the coordinates during mapping for the initialisation of the coarse prediction coordinates.

APM-MPAPT fine-tuned the existing model weights of LoFTR [1], which is trained on MegaDepth [19]. Simultaneously, we freeze the weights of the CNN Backbone, preventing them from changing during the training process. With this pre-trained weights, we just need to select only a small subset of frames from one video. Specifically, for one video, we sorted the frames based on the number of point annotations in ascending order. We select the first 30% to form a non-repeating combination of frame pairs, randomly choosing only 200 pairs from them, resize them to 256 × 256 resolution. So for the 28 training set videos, we utilised only 5600 frames in total. Considering that the training set contains a large number of static points, and to enhance the model's ability to track dynamic points, during training, we apply a slight random global movement to the two frames to be matched, akin to applying a subtle jitter.

When fine-tuning, we set the learning rate of 3×10^{-5}, the weights in loss are as follows: $w_{cm} = 0.1$, $w_{cou} = 0.8$, $w_f = 1.0$.

4 Experiments

4.1 Quantitative Comparison

We evaluate APM-MPAPT on validation set in point tracking part of the Perception Test [3] benchmark. Follow Perception Test, the model processes images at a 256 × 256 resolution, without maintaining the aspect ratio. The evaluation follows the same metrics: (1) $Position\ Accuracy(< \delta^x)$: given a threshold δ we compute the proportion of points that fall within this threshold of their ground truth locations in frames where the points are visible. Predictions are resized to 256 × 256 resolution, and we measure accuracy at five thresholds: 1, 2, 4, 8, and 16 pixels. (2) $Occlusion\ Accuracy(OA)$: a straightforward classification accuracy for predicting whether a point is occluded in each frame. (3) $Jaccard\ at\ \delta$: an evaluation metric accounts for both occlusion and position accuracy. It represents the fraction of 'true positives', i.e., points within the threshold of any visible ground truth points, divided by the sum of 'true positives', 'false positives'(points predicted as visible but are either occluded or farther than the threshold), and 'false negatives'(visible ground truth points predicted as occluded or where the prediction is farther than the threshold). The final metric, $AverageJaccard(AJ)$, averages Jaccard across all 5 thresholds: 1, 2, 4, 8, and 16 pixels.

LoFTR*. We also evaluated the performance of the original LoFTR [1] model on this task. LoFTR was designed for keypoint matching, but its dense matching mechanism can also be leveraged to predict designated points without occlusion prediction. Similar to our proposed method, we change LoFTR to predict by finding the maximum value in the designated row (located by query points) of the confidence matrix P_c. It is important to note that in the original model, fine-level prediction involves mapping integer coordinates from the coarse to fine level. Even though we retain the decimal part of the query point during the matching at the coarse level, the final prediction results still have a considerable

Table 1. Evaluation on the validation set of point tracking task in Perception Test [3] benchmark. **Top:** Average Jaccard (AJ), higher is better. **Middle:** Occlusion Accuracy (OA), Position Accuracy ($< \delta^x$) and its average value ($< \delta^x_{avg}$), higher is better. **Bottom:** Jaccard at δ, higher is better. Bold indicates the best performance, and underline indicates the second-best performance.

Method	**All points**	**static camera**	**moving camera**
Static Baseline [3]	0.384	0.436	0.094
LoFTR* [1]	0.447	0.475	0.290
TAP-Net [20]	**0.511**	0.530	**0.400**
APM-MPAPT	0.509	**0.535**	0.358

Method	**OA**	$< \delta^x_{avg}$	$< \delta^0$	$< \delta^1$	$< \delta^2$	$< \delta^3$	$< \delta^4$
Static Baseline [3]	0.733	0.597	0.394	0.511	0.601	0.695	0.784
LoFTR* [1]	0.733	**0.703**	**0.436**	**0.636**	**0.760**	**0.820**	**0.861**
TAP-Net [20]	**0.850**	0.624	0.217	0.517	0.740	0.811	0.835
APM-MPAPT	0.781	0.637	0.374	0.591	0.706	0.747	0.768

Method	**Jac.** δ^0	**Jac.** δ^1	**Jac.** δ^2	**Jac.** δ^3	**Jac.** δ^4
Static Baseline [3]	0.228	0.318	0.387	0.457	0.530
LoFTR* [1]	0.239	0.384	0.488	0.542	0.582
TAP-Net [20]	0.126	0.366	**0.613**	**0.707**	**0.741**
APM-MPAPT	**0.245**	**0.444**	0.575	0.626	0.653

margin of error. The changed LoFTR with zero-shot denoted as LoFTR* is also compared to the other method in Table 1.

APM-MPAPT. The processing of two frames with APM-MPAPT has been thoroughly described in the third section. For long-term tracking within a video, we simultaneously predict all the query points on the query frame which is the first frame of query points, by conducting matching the query frame with every subsequent frame in one video sequence. Table 1 shows the results. We also exhibit some prediction examples on frames from the validation set of the Perception Test [3] in Fig. 3. APM-MPAPT successfully predicts significant occlusions, correcting many erroneous coordinates predicted by the LoFTR* model. At the same time, it ensures a high level of position accuracy. Although APM-MPAPT performs well in areas with weak textures, its capability to recognise objects is relatively poor. This is evident when similar objects exchange positions, as APM-MPAPT may fail to accurately identify them.

Quantitative Analysis. To further analyse the experimental results, following Perception Test [3], we divide the video into static camera scenario and moving

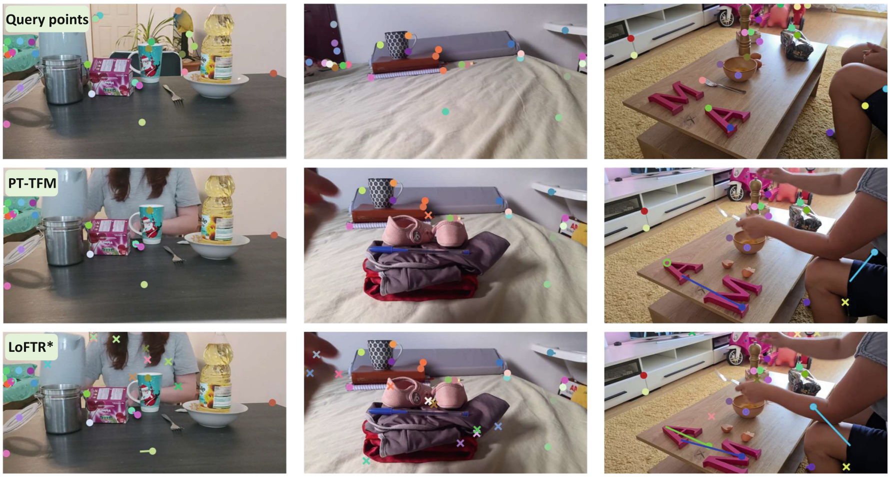


Fig. 3. APM-MPAPT compared to LoFTR*. Top row represents the query points on the first frame of the videos. Predictions of both models for a subsequent frame are below. The filled circles represent our predicted points, with their connected ends being the ground truth(GT). X's indicate predictions where the GT is occluded, while empty circles denote location of the points which are visible in the GT but predicted as occluded (note: we did not retain the coordinates predicted as occluded). APM-MPAPT successfully predicts and eliminates a significant numbers of occlusions that LoFTR* fails to predict, while maintaining high position accuracy, even in texture-less areas such as surfaces of table or blanket. For similarly structured objects that switch positions, APM-MPAPT tends to predict erroneous coordinates as occluded.

camera scenario. For AJ metric, Due to the presence of numerous static points in the video, the static baseline also performs well to a certain extent. The performance of LoFTR*, which was modified slightly from LoFTR [1], has a significant improvement compared to the Static Baseline [3]. APM-MPAPT, after fine-tuning, has shown significant performance improvements across two distinct scenarios. Additionally, with our designed occlusion prediction, APM-MPAPT shows improvement in the OA metric compared to LoFTR*. Here, it is important to note that, for the evaluation of position accuracy, we only focus on predicting visible points. Although the model predicts coordinates for all points, including those predicted to be occluded, we do not retain the coordinates for these predicted occlusion points. Therefore, it is possible that erroneous occlusion prediction may result in lower position accuracy compared to the LoFTR*. Finally, APM-MPAPT outperforms both the static baseline and LoFTR* in Jaccard at all thresholds (δ^x). Compared to TAP-Net [20], which is trained on the large-scale synthetic Kubric dataset [25], APM-MPAPT employs a fine-tuning paradigm that uses only a small amount of training data (200 frames per video × 28 videos, totalling 6400 frames) and a short fine-tuning time (<2 h). With a much lower cost, APM-MPAPT achieves comparable performance as TAP-Net.

Table 2. Model ablation by removing specific component at a time. In "W/O Occlusion Estimate", we removed the losses related to occlusion and uncertainty from the model. After fine-tuning, we conducted testing without setting a visibility threshold, rather than simply omitting the visibility threshold during testing.

Average Jaccard (AJ)	All points	static camera	moving camera
Full Model	**0.509**	**0.535**	0.358
Agent Attention (n = 256)	0.509	0.534	0.365
Linear Attention	0.508	0.533	**0.367**
W/O Uncertainty Estimate	0.504	0.531	0.354
W/O Occlusion Estimate	0.440	0.462	0.307
W/O Fine-level Prediction	0.370	0.399	0.209

Table 3. The impact of different weights w_o during occlusion supervision. In both Eqs. (5) and (11), the same weights are used for supervising occlusion predication.

Average Jaccard (AJ)	All points	static camera	moving camera
$w_o = 1.00$(Full Model)	**0.509**	**0.535**	0.358
$w_o = 0.75$	0.502	0.527	0.363
$w_o = 0.50$	0.502	0.526	**0.365**
$w_o = 0.25$	0.496	0.521	0.353

Furthermore, due to the high precision requirements in LoFTR's application scenarios, APM-MPAPT outperforms TAP-Net at stricter thresholds after fine-tuning.

4.2 Ablation Studies

We evaluate the impact of different design in APM-MPAPT, and the results are shown in Table 2. The result can evident that the fine matching stage and occlusion prediction play crucial roles in the model's performance. In dynamic scenes particularly, the fine-grained prediction by Motion-driven Point Association (MPA) significantly enhances the model's accuracy. Similarly, the improvement in occlusion prediction also indicates the robustness of APM-MPAPT against most occlusion scenarios. Even when dealing with points that reappear after prolonged occlusion over time, APM-MPAPT demonstrates sufficient performance. The introduction of uncertainty also slightly improves the model's performance. We also compared the impact of different attention mechanisms on model performance. Since LoFTR [1], the pre-trained model, originally uses linear attention, which is well-suited to the existing parameters and performs well in point tracking tasks. Additionally, we tested the impact of increasing the size of the Agent (n) in Agent Attention, which yielded better performance in dynamic scenes. Through comparison, we ultimately chose to use Agent Attention ($n = 49$), which is a form of generalised Linear Attention, for the full model.

Considering the significant improvement in model performance with the introduction of occlusion prediction, we further explored the impact of the supervision weight on occlusion prediction, and the results are shown in Table 3. Specifically, we applied weighting to the supervision of occlusion in Eqs. (5) and (11), i.e., $w_o * \mathrm{BCE}(\hat{o}, o)$ and $w_o * \mathrm{BCE}(\hat{u}, o)$, with weights w_o set at 0.25, 0.50, and 0.75. Through comparison, our final model sets $w_o = 1.00$.

5 Conclusion

In this paper, we introduce APM-MPAPT, which utilises an attention-based patch feature matching method associated with motion information to accomplish the task of point tracking. From a global rough prediction initialisation to a local refinement, a coarse-to-fine tracking framework is formed. APM-MPAPT offers a more robust matching approach for the task of point tracking, maintaining excellent accuracy when dealing with two frames of arbitrary temporal distance. Although the matching approach for arbitrary two frames enables ultra-long-term video tracking, it still exhibits sub-optimal performance in maintaining the temporal continuity of point trajectories. Nevertheless, our method broadens the scope of solutions for point tracking tasks.

Acknowledgments. This work was supported in part by the National Key R&D Program of China (2023YFE0116300).

This work was supported in part by the National Natural Science Foundation of China (62106089, 62336004, 62020106012, 62332008).

References

1. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: LoFTR: detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8922–8931 (2021)
2. Doersch, C., et al.: Tapir: tracking any point with per-frame initialization and temporal refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10061–10072 (2023)
3. Patraucean, Vet al.: Perception test: a diagnostic benchmark for multimodal video models. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
4. Ma, J., Jiang, X., Fan, A., Jiang, J., Yan, J.: Image matching from handcrafted to deep features: a survey. Int. J. Comput. Vis. **129**(1), 23–79 (2021)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
6. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9, pp. 404–417. Springer (2006)
7. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)

8. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: learned invariant feature transform. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14, pp. 467–483. Springer (2016)
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 224–236 (2018)
10. Manuelli, L., Li, Y., Florence, P., Tedrake, R.: Keypoints into the future: self-supervised correspondence in model-based reinforcement learning. arXiv preprint arXiv:2009.05085 (2020)
11. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
12. Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., Sivic, J.: Neighbourhood consensus networks. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
13. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
14. Jiang, W., Trulls, E., Hosang, J., Tagliasacchi, A., Yi, K.M.: COTR: correspondence transformer for matching across images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6207–6217 (2021)
15. Yu, J., Chang, J., He, J., Zhang, T., Yu, J., Wu, F.: Adaptive spot-guided transformer for consistent local feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21898–21908 (2023)
16. Huang, D., Chen, Y., Liu, Y., Liu, J., Xu, S., Wu, W., Ding, Y., Tang, F., Wang, C.: Adaptive assignment for geometry aware local feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5425–5434 (2023)
17. Zhu, X.F., Xu, T., Wu, X.J., Kittler, J.: Feature enhancement and coarse-to-fine detection for RGB-D tracking. Pattern Recogn. Lett. (2024)
18. Xu, T., Kang, Z., Zhu, X., Wu, X.J.: Learning adaptive spatio-temporal inference transformer for coarse-to-fine animal visual tracking: algorithm and benchmark. Int. J. Comput. Vis. 1–15 (2024)
19. Li, Z., Snavely, N.: MegaDepth: learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2041–2050 (2018)
20. Doersch, C., et al.: Tap-vid: a benchmark for tracking any point in a video. Adv. Neural. Inf. Process. Syst. **35**, 13610–13626 (2022)
21. Harley, A.W., Fang, Z., Fragkiadaki, K.: Particle video revisited: tracking through occlusions using point trajectories. In: European Conference on Computer Vision, pp. 59–75. Springer (2022)
22. Lin, T.Y., Dollár, 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, pp. 2117–2125 (2017)
23. Han, D., Ye, T., Han, Y., Xia, Z., Song, S., Huang, G.: Agent attention: on the integration of softmax and linear attention. arXiv preprint arXiv:2312.08874 (2023)
24. 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)
25. Greff, K., et al.: Kubric: a scalable dataset generator. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3749–3761 (2022)

26. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4938–4947 (2020)
27. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: Lightglue: local feature matching at light speed. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17627–17638 (2023)
28. Xu, T., Wu, X.J., Kittler, J.: Non-negative subspace representation learning scheme for correlation filter based tracking. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 1888–1893. IEEE (2018)
29. Fan, H., et al.: Visdrone-sot2020: the vision meets drone single object tracking challenge results. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, pp. 728–749. Springer, Cham (2020)
30. Wen, J., Chu, H., Lai, Z., Xu, T., Shen, L.: Enhanced robust spatial feature selection and correlation filter learning for UAV tracking. Neural Netw. **161**, 39–54 (2023)
31. Xu, T., Zhu, X.F., Wu, X.J.: Learning spatio-temporal discriminative model for affine subspace based visual object tracking. Vis. Intell. **1**(1), 4 (2023)
32. Zhao, J., et al.: The 3rd anti-UAV workshop & challenge: methods and results. arXiv preprint arXiv:2305.07290 (2023)

SITUATE: Indoor Human Trajectory Prediction Through Geometric Features and Self-supervised Vision Representation

Luigi Capogrosso[1](✉), Andrea Toaiari[1], Andrea Avogaro[1], Uzair Khan[1],
Aditya Jivoji[2], Franco Fummi[1], and Marco Cristani[1]

[1] Department of Engineering for Innovation Medicine,
University of Verona, Verona, Italy
{luigi.capogrosso,andrea.toaiari,andrea.avogaro,uzair.khan,
franco.fummi,marco.cristani}@univr.it
[2] Visvesvaraya National Institute of Technology, Nagpur, India
adjivoji@students.vnit.ac.in

Abstract. Patterns of human motion in outdoor and indoor environments are substantially different due to the scope of the environment and the typical intentions of people therein. While outdoor trajectory forecasting has received significant attention, indoor forecasting is still an underexplored research area. This paper proposes SITUATE, a novel approach to cope with indoor human trajectory prediction by leveraging equivariant and invariant geometric features and a self-supervised vision representation. The geometric learning modules model the intrinsic symmetries and human movements inherent in indoor spaces. This concept becomes particularly important because self-loops at various scales and rapid direction changes often characterize indoor trajectories. On the other hand, the vision representation module is used to acquire spatial-semantic information about the environment to predict users' future locations more accurately. We evaluate our method through comprehensive experiments on the two most famous indoor trajectory forecasting datasets, *i.e.*, THÖR and Supermarket, obtaining state-of-the-art performance. Furthermore, we also achieve competitive results in outdoor scenarios, showing that indoor-oriented forecasting models generalize better than outdoor-oriented ones. The source code is available at https://github.com/intelligolabs/SITUATE.

Keywords: Human Trajectory Prediction · Geometric Deep Learning · Self-Supervised Vision Representation

1 Introduction

Human trajectory prediction is the task of predicting the likely path that a subject will take to reach its designated endpoint [30]. This predictive process finds its applicability and utility in a multitude of domains [22]. For example,

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 364–379, 2025.
https://doi.org/10.1007/978-3-031-78444-6_24

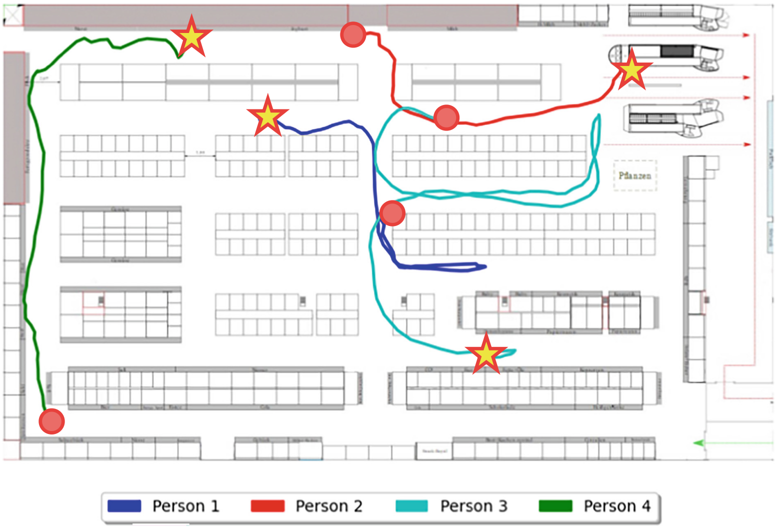


Fig. 1. Examples of different trajectories from the Supermarket [11] dataset to show the difficulty of the indoor trajectory prediction task. In particular, the dataset showcases long trajectories (Person 4), self-loops (Person 1 and Person 3), and confusing movements (Person 2) performed in an environment that strongly affects the people's paths. Specifically, the red circle represents the starting point of a trajectory, and the yellow star represents its final point. (Color figure online)

in the context of robotics, it serves as a tool for facilitating the predictions on potential future robot trajectories, useful for intelligent planning considering human responses [31]. In industry, human trajectory prediction becomes critical for optimizing automated systems and ensuring seamless interactions with other occupants and components of a production line [32].

Despite the significant volume of research over the past decade devoted to outdoor trajectory prediction [1,4,12,13,15,31,41], there has been a notable scarcity of studies that exploited user trajectory data in indoor settings [26,28,29,35,36,38,39], also considering the crucial role these predictions play nowadays in the development of location-based services within indoor spaces. This gap in research inspired this work, which investigates a learning framework designed explicitly for indoor trajectory prediction.

Motivations for This Paper. In Fig. 1, we can note the distinctive nature of indoor settings, where users can encounter numerous choices and potential pathways. This factor implies that the dynamic of the motion can be strongly influenced by the environment setup [6,26]. Users can navigate through different interconnected rooms, corridors, doors, and elevators, often having the freedom to deviate from straightforward paths and choose alternative routes. Indoor spaces also have a higher density of structural elements and potential obstacles, such as furniture, walls, and partitions, as shown in Fig. 1, related to the Supermarket [11] dataset. Outdoor environments provide more open spaces, where visibility is less restricted, and the impact of physical barriers is typically reduced [7].

Consequently, indoor trajectory prediction requires a deeper understanding of the context and semantics of the indoor space, as users may have specific goals, like finding a particular room, reaching a specific point of interest, or accessing various facilities [14]. This contextual richness adds a layer of complexity to the prediction process since it also makes it necessary to consider the space's physical layout. Considering the omnipresence of indoor environments in human lives, it is imperative to address trajectory forecasting in these situations. Indeed, recent studies show that humans spend most of their time in indoor environments such as homes, supermarkets, airports, conference facilities, and train stations [2]. These considerations form the basis of the research conducted in this work.

Innovations in This Paper. While outdoor trajectory forecasting has received significant attention, indoor forecasting is still an underexplored research area. As a result, we present SITUATE, the first model designed specifically to cope with indoor trajectory forecasting by leveraging equivariant and invariant geometric feature learning and a self-supervised vision representation. Taking inspiration from [42], the equivariant and invariant geometric learning modules were employed to accurately represent intrinsic movements, like self-loops at various scales and hierarchies inherent in indoor spaces. On the other hand, the self-supervised vision representation module enabled us to acquire spatial-semantic information about the environment, using the scene or space layout images when available, to predict users' future locations meaningfully and accurately.

In summary, the main contributions of this paper are:

- We present SITUATE, a novel approach for indoor human trajectory forecasting based on equivariant and invariant geometric feature learning modules and a self-supervised vision representation;
- The equivariant and invariant modules are used to cope with the problem related to the more complicated movements inherent in indoor spaces;
- The vision representation module is used to acquire spatial-semantic information about the environment to predict users' future locations more accurately;
- SITUATE also achieves competitive results in outdoor scenarios, showing that indoor forecasting models generalize better than outdoor-oriented ones.

2 Related Work

Indoor Human Trajectory Prediction. Predicting the evolution of a pedestrian trajectory in the future is a long-standing task whose interest is constantly renewed by the emergence of new scenarios that can benefit from it, *e.g.*, autonomous driving [30]. When proposing a methodology to tackle trajectory forecasting, one should take care of several aspects, from the environment's geometry [26] to the presence of obstacles [16] and the possible interactions between multiple agents [3]. Some traditional methods to approach this task involved force models [17], Markov models [21], and RNNs [46]. Notably, considering common sense rules and conventions that humans observe in social spaces helps to manage simultaneous predictions in crowded scenes [1].

Multiple deep learning-based models have been applied successfully to forecast pedestrian trajectories, such as GNNs [44], Trasformers [12] and Conditional Variational AutoEncoders (CVAEs) [25]. More recently, diffusion models have also been applied to solve this problem [13]. However, most proposed methods are tested only on datasets representing outdoor scenarios. This is due to a lack of comprehensive indoor datasets and the fact that indoor trajectories can be considered more "difficult" or non-linear [28]. When traversing indoors, our immediate movement decision is influenced by the objects in our path and the surrounding walls [26]. In indoor settings, people navigate in loosely constrained but cluttered spaces with multiple goal points that can be reached in many ways [29]. Moreover, people in indoor scenarios tend to focus on their surroundings, fixating on the most interesting parts of the scene, alternating movement and stationary phases [36]. At the same time, outside, the movement area can be much larger, and the subjects can move further apart.

Some works have been proposed to address the specific problem of indoor trajectory forecasting [26,28,29,35,39], highlighting the differences between indoor and outdoor trajectory forecasting. In [28], the authors address the problem of generalizability, proposing a novel indoor dataset and new metrics to normalize common biases. They tackle the problem of aleatoric multimodality with the GAN-Tri model, which uses a heuristic to produce samples corresponding to different behaviors. [26] examine trajectories, modeled as a Markov chain, within 3D environments, introducing the concept of an occupancy map to represent the relative accessibility of each point on the map with respect to its geometry. The study emphasizes the importance of proximity from each point to the destination and the occupancy frequency in constructing a probability transition matrix for trajectory prediction. Unlike them, our approach considers indoor spaces' detailed scene layouts and non-trivial human movements.

Equivariant and Invariant Graph Neural Networks. Inspired by the research on rotation-equivariant convolutional neural networks within the 2D image domain [8], the advent of Graph Neural Network (GNN) architectures opened doors to investigating symmetries beyond rotations [43]. For example, in [33], the authors proposed partial equivariance by focusing on translation equivariance. Meanwhile, [10] constructed filters using spherical harmonics, enabling equivariance to rotations and translations and facilitating transformations between higher-order representations.

In [34], a new model for learning equivariant graph neural networks, dubbed EGNNs, is proposed. Differently from the previous works, this formulation maintains the flexibility of GNNs while remaining E(n) equivariant (translation, rotation, and reflection equivariant) without the need to compute expensive higher-order operations. [19] further extended this concept by incorporating geometrical constraints implicitly encoded in the forward kinematics when tackling molecular dynamics prediction and human motion capture. However, a significant limitation of current methods is their focus solely on state prediction, preventing models from effectively using sequence information.

Recently, EqMotion [42] extended on these ideas to propose an equivariant motion prediction parametric network with an invariant interaction reasoning module, able to tackle distinct problems such as particle and molecule dynamics, human pose forecasting, and outdoor pedestrian trajectory prediction. Interaction invariance is fundamental in ensuring the agents' interactions remain constant under input transformation.

In our research, we adapt some of the concepts presented in [42] to propose an equivariant model for the human trajectory prediction task, which, in combination with a module to extract semantic information about the scenes, unlock more precise forecasting capabilities in indoor settings.

Self-supervised Vision Representation. One way to get image representations without heavily relying on annotated data is to perform Self-Supervised Learning (SSL). In a nutshell, SSL learns deep feature representations invariant to sensible transformations of the input data. Then, the learned representations could be used in supervised downstream tasks.

The self-supervised vision representation state-of-the-art rapidly evolved, with Transformer-based architectures emerging as leading models. The Vision Transformer (ViT) [9], and its variants like DeiT [37], have demonstrated impressive performance in learning powerful visual representations from unlabeled data. Specifically, these models leverage self-attention mechanisms to capture global context and long-range dependencies within images, enabling them to encode rich semantic information efficiently.

In this paper, to extract semantic information from scenes represented in a 2D map, we use the pre-trained BEiT [5], the state-of-the-art self-supervised vision representation model. This offers a powerful framework for learning visual representations without explicit supervision, effectively capturing high-level semantics and intricate features inherent in visual data.

3 Method

Mathematical Background. Given a set of transformations $T_x : X \rightarrow X$, a function $F : X \rightarrow Y$ is called Equivariant if exists a transformation $T_y : Y \rightarrow Y$ equivalent to T_x, on the Euclidean space such that:

$$F(T_x(X)) = T_y(F(X)) \ . \tag{1}$$

Moreover, we also want the model to have the invariance property. Given the same set of transformations, a function $F : X \rightarrow Y$ is called Invariant on the Euclidean space if it exists a transformation $T_y : Y \rightarrow Y$ such that:

$$F(X) = F(T_x(X)) \ . \tag{2}$$

Specifically, this work addresses the problem of multi-person trajectory forecasting by considering the input trajectories as a graph. As proven by [34], during the message passing of a GNN, the property of equivariance can be ensured by enriching the features of the neighbor nodes with the $L2$ distance between nodes. Let $G = \{V, E\}$ be an input graph representing the input trajectory with nodes $v_i \in V$ and edges $e_{ij} \in E$. For every node v_i, a feature vector $h \in \mathbf{R}^h$ and an absolute position $x_i \in \mathbf{R}^3$ are given. To preserve equivariance among different layers of the model, we update the position as follows:

$$m_{ij} = \phi_e \left(h_i^l, h_j^l, \left\| x_i^l - x_j^l \right\|^2 \right) \ , \tag{3}$$

$$x_i^{l+1} = x_i^l + C \sum_{j \neq i} \left(x_i^l - x_j^l \right) \phi_x \left(m_{ij} \right) \ , \tag{4}$$

where C is equal to $1/(M-1)$ with M number of nodes, ϕ_e and ϕ_x are learnable Multi-layer Perceptrons (MLPs), defined as $\phi_e(\cdot) = W_e \cdot + B_e$, l indicates the layer and m_{ij} represents the information passed between two nodes during the message passing.

As reported in [34], ϕ_x has to be a scoring function $\phi_x : X \rightarrow S$, with $S \in \mathbf{R}^1$. With this procedure, the update of V is consistent, allowing the model to learn without being affected by SO(2) transformations, with SO(2) being the group of all rotations in the plane around the origin that preserve the Euclidean norm, mathematically described by 2 × 2 matrices. Furthermore, the features learned across layers must be consistent and invariant to graph transformations. To do so, the following procedure governs the final message-passing operations and the update of the features carried out by the $i-th$ layer:

$$m_i = \sum_{j \neq i} m_{ij} \ , \tag{5}$$

$$h_i^{l+1} = \phi_h \left(h_i^l, m_i \right) \ , \tag{6}$$

with ϕ_h, an MLP also designed as $\phi_h(X) = W_h X + B_h$, responsible for the invariant feature learning. Mixing these two components allows us to build an Equivariant and Invariant GNN using Euclidean SO(2) transformations.

Motion Prediction. Here, we introduce the general problem formulation of motion prediction. We have a multi-agent system with m agents. Each agent is represented as A_i, where $i = 1, 2, \ldots, m$. The goal is to predict the future motions of these agents based on their historical observations. For each agent A_i, we can denote the historical observations as X_i. These observations typically include positions and can be represented as $X_i = \{x_0^i, x_1^i, \ldots, x_t^i\}$, where x_t^i represents the position of agent A_i at time step t. We also add the velocity $S_i = \{s_{t+1}^i, s_{t+2}^i, \ldots, s_{t+f}^i\}$ as input information of the model. The velocity of an agent is a natural invariant feature because it is not affected by any translation or SO(2) transformation. We use the velocity to compute the initial feature vector of a specific agent A_i. More details in Sect. 3.2. Specifically, for each agent A_i we aim to predict its future f positions $Y_i = \{y_{t+1}^i, y_{t+2}^i, \ldots, y_{t+f}^i\}$.

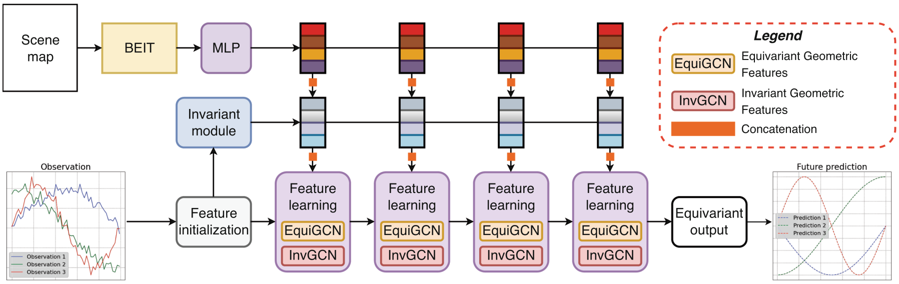


Fig. 2. In SITUATE, we first produce a feature vector regarding the scene using the self-supervised vision representation module. Then, a feature initialization layer is used to initialize geometric and pattern features. We then successively update the geometric and pattern features by the equivariant geometric feature learning and invariant pattern feature learning layers, obtaining expressive feature representation. We further use an invariant reasoning module to infer an interaction graph used in equivariant geometric feature learning. Finally, we use an equivariant output layer to obtain the final prediction.

3.1 The SITUATE Prediction Network

In this section, we present SITUATE, our motion prediction network that explicitly uses equivariant and invariant geometric features and a self-supervised scene representation module to tackle the indoor trajectory prediction problem. The model architecture is shown in Fig. 2.

The first module we present is in charge of producing the scene-representation encoding. As anticipated, the subjects' motion characteristics differ greatly from those of the outdoor case when considering indoor trajectory forecasting. Indeed, the motion is strongly characterized and limited by the objects and obstacles in the scene. Knowing the available space that limits the viable paths in the scene can, for every X_i, strongly reduce the cardinality of all the possible outcomes of the model. Starting from the assumption that all the scene objects and structure are available in the form of a scene layout or a camera image, BEiT [5] is first used to output visual tokens T_s, the so-called scene-representation encodings. These tokens T_s are fed into a learnable MLP defined as $\phi_t : T_s \rightarrow T_e$ and then concatenated into the input.

The input concatenated with T_e is then fed into two modules: Equivariant block (*EquiGCN*) and Invariant block (*InvGCN*). Following [42], these two blocks are both based on the implementation of the message passing described in Eq. 3, modified to accept also T_e:

$$m_{ij} = \phi_e \left(h_i^l, h_j^l, \left\| x_i^l - x_j^l \right\|^2, T_e, a_{ij} \right) , \tag{7}$$

where a_{ij} is the edge attribute (or weight), which can be derived from the adjacency matrix.

Specifically, the $EquiGCN$ block is responsible for the update of the node's coordinates x, and it represents the implementation of the update function described in Eq. 4. On the other hand, *InvGCN* is the implementation of the update function of the node's features h in Eq. 6.

The possible pathways are learned by the module ϕ_t, starting from the token produced by the pre-trained BeIT model. The outputs of *EquiGCN* and *InvGCN* are computed as reported respectively in Eq. 4 and Eq. 6, updating h_i^{l+1} and x_i^{l+1}.

To understand the contribution of these two modules in less formal and more practical terms, imagine a navigation system. It can get from point A to point B but might struggle with tricky situations. In SITUATE, EquiGCN injects a sense of direction like a compass you wear on your hat. Specifically, it ensures the network understands the environment's layout, regardless of where it starts "looking". InvGCN, on the other hand, acts like a map you hold - it helps the network account for different starting points and body orientations, making the predictions more robust.

3.2 Feature Initialization

The input given to our model is a set of trajectories of different agents. The first step is to define a node for every position x_t for every agent A_i. Every node x_t is connected with x_{t-1} and x_{t+1} if they are related to the same agent A_i. Since only trajectories (and thus positions x_t) are given as starting data, it is necessary to define for each trajectory a vector of initial features h_i^0 to be used as input together with the positions x_i^0.

As stated in [19], having an invariant feature vector h_i^0 is necessary to guarantee equivariance. Given that as input data we only have position X_i, we followed the procedure in [42] to use velocities in order to create h_i^0 as follows:

$$\hat{x}_i = \phi_X(X_i + \overline{\mathbb{H}}) + \overline{\mathbb{H}} \ , \tag{8}$$

$$\rho_i^t = \|v_i^t\|_2 \ , \tag{9}$$

$$\theta_i^t = \text{angle}(v_i^t, v_i^{t-1}) \ , \tag{10}$$

$$h_i^0 = \phi_{h_0}(\rho_i, \theta_i) \ , \tag{11}$$

where h_i^0 is the initial features vector of the $i-th$ agent. v_i^t represent the velocity of the agent and is defined as $\triangle\hat{x}_i^t$, where $\triangle$ is the finite difference operator, $\overline{\mathbb{H}}$ is the centroid of the observed trajectories of all agents in the scene. ϕ_X and ϕ_{h_0} are two fully connected layers responsible for encoding and producing the initial graph and the initial features of the trajectory.

To compute h_i^0, two different types of velocities are needed (thus, information invariant to rotation and translation): $\triangle x_i^t$ effectively represents the Euclidean velocity of the agent and θ_i^t represents the angular velocity on a certain time step t. Note that both ϕ_{x_0} and ϕ_{h_0}, and in general all the operations described, are linear transformations: this is necessary to preserve both equivariance and invariance properties of the remaining part of the model.

4 Experiments

Our experimental evaluation is tailored toward two objectives. Firstly, in Sect. 4.2, we show the superiority of SITUATE in the two most well-known indoor datasets, defining the new state-of-the-art in indoor scenarios. Secondly, in Sect. 4.3, we prove that SITUATE can also achieve comparable results with respect to other competitors on outdoor datasets. Finally, in Sect. 4.4, we report some ablation studies.

4.1 Evaluation Setup

Datasets. We evaluate SITUATE on the state-of-the-art indoor datasets and the most well-known outdoor human trajectory prediction dataset.

THÖR. The THÖR dataset [29] includes human motion trajectory and gaze data collected in an indoor environment with accurate ground truth for the participants' position. It comprises 395K frames at 100 Hz, 2531K people detections, and over 600 individual and group trajectories between multiple resting points. The map was taken from the dataset's official website.

Supermarket. The Supermarket dataset [11] comprises 4 different scenario: German1, German2, German3, and German4, *i.e.*, four different supermarket. The dataset collection involved attaching devices on shopping carts/baskets and recording their movements during customer usage. Each subset includes a file with a map of the supermarket.

ETH-UCY. The ETH [27] and UCY [23] dataset group consists of five different scenes: ETH & HOTEL (from ETH) and UNIV, ZARA1, &ZARA2 (from UCY). The scenes are captured in unconstrained outdoor environments with few objects blocking the pedestrian paths. In this case, images of the scene were used.

Evaluation Metrics. We use standard metrics for the trajectory prediction task, *i.e.*, minimum Average Displacement Error (ADE), and minimum Final Displacement Error (FDE). In particular, ADE measures the average $L2$ difference between the prediction at all time steps and the ground truth. On the other hand, FDE measures the difference between the predicted endpoint and the ground truth.

Prediction Mode. Following the evaluation protocol of [42], SITUATE is employed in two prediction modes: deterministic and multi-prediction. Deterministic means the model only outputs a single prediction for each input motion observation, while multi-prediction means the model has 20 predictions for each input motion observation. Under multi-prediction, ADE and FDE will be calculated using the best-predicted trajectory. To adapt to multi-prediction, we modify SITUATE to repeat the last feature updating layer and the output layer 20 times in parallel to have a multi-head prediction.

Implementation Details. As a backbone for our model, we used the structure of [42]. The model architecture has four layers of geometric feature learning.

Table 1. Deterministic prediction performance (ADE (m)/FDE (m)) on the THÖR and the Supermarket datasets. The **bold**/underlined font denotes the **best**/second-best result.

	Performance (ADE (m) ↓ / FDE (m) ↓)		
Deterministic Evaluation	THÖR	Supermarket	Average
TransF [12]	2.62/4.81	2.56/2.90	2.59/3.85
MemoNet [41]	0.78/5.05	1.79/2.94	1.28/3.99
EqMotion [42]	0.56/0.94	1.71/2.65	1.13/1.79
SITUATE (ours)	**0.45/0.93**	**1.21/1.84**	**0.83/1.38**

Table 2. Multi-prediction performance (ADE (m)/FDE (m)) on the THÖR and the Supermarket datasets. The **bold**/underlined font denotes the **best**/second-best result.

	Performance (ADE (m) ↓ / FDE (m) ↓)		
Multi-prediction Evaluation	THÖR	Supermarket	Average
PECNet [25]	–	1.57/3.45	–
GP-Graph [4]	2.80/3.92	3.19/4.57	2.99/4.24
EqMotion [42]	1.32/1.03	1.29/1.77	2.61/1.40
SITUATE (ours)	**0.50/0.86**	**0.53/0.65**	**0.51/0.75**

We use the SiLU activation function and dropout with a 0.5 probability to regularise within all MLPs. The visual embeddings of the image, *i.e.*, the floor plans (look at Fig. 1) for context information are derived from the last layer of the BEiT model. The model is provided with past trajectory information spanning eight discrete time steps, and the model's task is to predict 12 steps into the future. In addition to the dropout mentioned above, we apply the Discrete Cosine Transform (DCT) to the input data as a regularisation technique. Specifically, by representing the data in the frequency domain, it becomes easier to distinguish between signal and noise components, resulting in a cleaner signal. The impact of these regularization approaches is discussed in Sect. 4.4. We train our models with a batch size of 64 for 60 epochs, using AdamW [24] as an optimizer within the PyTorch Lightning framework on an NVIDIA RTX 3090.

4.2 Indoor Human Trajectory Prediction Results

We conducted comparative experiments to assess the soundness of our approach against existing trajectory prediction methods. The methods include deterministic evaluation models (TransF [12], MemoNet [41]), as well as multi-prediction evaluation models (PECNet [25], GP-Graph [4]). Our evaluation for both prediction modes also encompasses EqMotion [42], the state-of-the-art method with invariant end equivariant interaction reasoning. Table 1 and Table 2 show the results.

Table 3. Deterministic prediction performance (ADE (m)/FDE (m)) on the ETH-UCY dataset. The **bold**/underlined font denotes the best/second-best result.

	Performance (ADE (m) ↓ / FDE (m) ↓)					
Deterministic	ETH	HOTEL	UNIV	ZARA1	ZARA2	Average
S-LSTM [1]	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
SGAN-ind [15]	1.13/2.21	1.01/2.18	0.60/1.28	0.42/0.91	0.52/1.11	0.74/1.54
Traj++ [31]	1.02/2.00	0.33/0.62	0.53/1.19	0.44/0.99	0.32/0.73	0.53/1.11
TransF [12]	1.03/2.10	0.36/0.71	0.53/1.32	0.44/1.00	0.34/0.76	0.54/1.17
MemoNet [41]	1.00/2.08	0.35/0.67	0.55/1.19	0.46/1.00	0.37/0.82	0.55/1.15
EqMotion [42]	0.96/1.92	**0.30**/0.58	**0.50/1.10**	**0.39/0.86**	**0.30/0.68**	**0.49/1.03**
SITUATE (ours)	**0.94/1.90**	**0.30/0.57**	**0.50/1.10**	0.41/0.89	0.32/0.70	**0.49/1.03**

Table 4. Multi-prediction performance (ADE (m)/FDE (m)) on the ETH-UCY dataset. The **bold**/underlined font denotes the best/second-best result.

	Performance (ADE (m) ↓ / FDE (m) ↓)					
Multi-prediction	ETH	HOTEL	UNIV	ZARA1	ZARA2	Average
SGAN [15]	0.87/1.62	0.67/1.37	0.76/0.52	0.35/0.68	0.42/0.84	0.61/1.21
STGAT [20]	0.65/1.12	0.35/0.66	0.34/0.69	0.29/0.60	0.52/1.10	0.43/0.83
STAR [44]	0.36/0.65	0.17/0.36	0.31/0.62	0.29/0.52	0.22/0.46	0.26/0.53
NMMP [18]	0.61/1.08	0.33/0.63	0.52/1.11	0.32/0.66	0.43/0.85	0.41/0.82
Traj++ [31]	0.61/1.02	0.19/0.28	0.30/0.54	0.24/0.42	0.18/0.31	0.30/0.51
PECNet [25]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
Agentformer [45]	0.45/0.75	0.14/0.22	0.25/0.45	0.18/**0.30**	0.14/0.24	0.23/0.39
GroupNet [40]	0.46/0.73	0.15/0.25	0.26/0.49	0.21/0.39	0.17/0.33	0.25/0.44
MID [13]	**0.39**/0.66	0.13/0.22	**0.22**/0.45	**0.17/0.30**	**0.13**/0.27	**0.21**/0.38
GP-Graph [4]	0.43/0.63	0.18/0.30	0.24/**0.42**	**0.17**/0.31	0.15/0.29	0.23/0.39
EqMotion [42]	0.40/**0.61**	**0.12/0.18**	0.23/0.43	0.18/0.32	**0.13/0.23**	**0.21/0.35**
SITUATE (ours)	0.41/0.64	0.13/0.20	0.23/0.43	0.22/0.35	0.14/0.26	0.22/0.37

The results show that the proposed SITUATE consistently outperforms all baseline methods in all cases. On the THÖR dataset, SITUATE achieves an ADE of 0.45 and an FDE of 0.93, showcasing its superiority over other models. Notably, compared to EqMotion, the second-best model, SITUATE exhibits a substantial 22% reduction in ADE and a 1% reduction in FDE.

Similarly, on the Supermarket dataset, SITUATE continues demonstrating its effectiveness with an ADE of 1.21 and an FDE of 1.84. Compared to EqMotion, again the closest competitor, SITUATE achieves a 29% reduction in ADE and a 31% in FDE, reinforcing its dominance.

Table 5. Ablation results (ADE (m)/FDE (m)) of SITUATE. We assess the contribution of the scene representation module and regularization methods in the deterministic prediction case.

		Performance (ADE (m) ↓ / FDE (m) ↓)		
Scene Representation	Regularization	THOR	Supermarket	Average
✗	✗	0.50/1.02	1.92/1.55	1.21/1.29
✗	✓	0.56/0.74	1.79/2.94	1.18/1.84
✓	✗	0.57/0.96	1.29/1.89	0.93/1.43
✓	✓	**0.45/0.93**	**1.21/1.84**	**0.83/1.38**

The consistently good performance of SITUATE across both datasets underscores its robustness and efficacy in trajectory prediction tasks. The results further suggest that SITUATE is accurate and that using scene information when tackling indoor prediction scenarios offers a key advantage compared to the available approaches.

4.3 Outdoor Human Trajectory Prediction Results

We also evaluate the performance of SITUATE with the deterministic and multi-prediction modalities with outdoor scenarios. Here, we show that SITUATE achieves competitive results, showing that indoor-oriented forecasting models tend to generalize better than outdoor-oriented ones. Table 3 and Table 4 present the quantitative results.

Specifically, when considering the deterministic prediction case, SITUATE demonstrates a performance improvement by obtaining state-of-the-art results in both ADE and FDE across the ETH (0.94/1.90), HOTEL(0.30/0.57), and UNIV (0.50/1.10) scenes. It places second in the ZARA1 and ZARA2 scenes while performing on par with EqMotion [42] when considering average performance. In the context of multi-prediction modality, SITUATE secures the second rank in terms of ADE and FDE across nearly 75% of the scenes within the ETH-UCY dataset while maintaining an overall second place in average performance.

Designed for indoor scenarios and their peculiar conformations, SITUATE remarkably demonstrates robust capabilities, even when tested on outdoor datasets. In contrast, this is not always true for architectures tailored for outdoor instances, as we can observe in Table 1 and Table 2, that often struggle when confronted with scenes that differ from those for which they were designed.

4.4 Ablation Study

We quantitatively evaluate the impact of the scene representation module and regularization methods by considering the deterministic indoor prediction scenario. Results are summarized in Table 5. It is observed that both contributions play a crucial role in enhancing the overall performance of SITUATE. In particular, the scene representation module effectively encodes semantic information from the visual scene maps, facilitating an accurate understanding of the environment. On the other hand, the regularization methods ensure robustness and generalization of the model by mitigating overfitting and improving its ability to generalize to unseen data. Therefore, we assert that both contributions are indispensable for achieving the desired outcomes and validating the efficacy of our approach.

5 Conclusion

This paper presents SITUATE, a graph neural network-based model designed specifically to cope with indoor human trajectory prediction. SITUATE, using geometric features and self-supervised vision representations, models the intricate human movements inherent in indoor spaces and accurately predicts users' future locations. The scene vision representation module provides insights about the environment, particularly helping in those indoor scenes that are more constrained and full of obstacles. We evaluate our method on two well-known indoor trajectory prediction datasets, *i.e.*, THÖR and Supermarket, and achieve state-of-the-art prediction performance. Furthermore, we also achieve competitive results in outdoor scenarios, showing that indoor-oriented forecasting models generalize better than outdoor-oriented ones.

Acknowledgement. This study was carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) - Missione 4 Componente 2, Investimento 1.5 - D.D. 1058 23/06/2022, ECS_00000043). This manuscript reflects only the Authors' views and opinions. Neither the European Union nor the European Commission can be considered responsible for them.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
2. ASHRAE, A.: Guideline 10p, interactions affecting the achievement of acceptable indoor environments (2010)
3. Aydemir, G., Akan, A.K., Güney, F.: Adapt: efficient multi-agent trajectory prediction with adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
4. Bae, I., Park, J.H., Jeon, H.G.: Learning pedestrian group representations for multi-modal trajectory prediction. In: European Conference on Computer Vision. Springer (2022)
5. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: bert pre-training of image transformers. In: International Conference on Learning Representations (2021)

6. Capogrosso, L., Skenderi, G., Girella, F., Fummi, F., Cristani, M.: Toward smart doors: a position paper. In: International Conference on Pattern Recognition. Springer (2022)
7. Choi, C., Dariush, B.: Looking to relations for future trajectory forecast. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
8. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International Conference on Machine Learning. PMLR (2016)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
10. Fuchs, F., Worrall, D., Fischer, V., Welling, M.: Se (3)-transformers: 3d roto-translation equivariant attention networks. In: Advances in Neural Information Processing Systems (2020)
11. Gabellini, P., DAloisio, M., Fabiani, M., Placidi, V.: A large scale trajectory dataset for shopper behaviour understanding. In: New Trends in Image Analysis and Processing–ICIAP 2019: ICIAP International Workshops, BioFor, PatReCH, e-BADLE, DeepRetail, and Industrial Session, Trento, Italy, September 9–10, 2019, Revised Selected Papers 20. Springer (2019)
12. Giuliari, F., Hasan, I., Cristani, M., Galasso, F.: Transformer networks for trajectory forecasting. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE (2021)
13. Gu, T., et al.: Stochastic trajectory prediction via motion indeterminacy diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
14. Guo, S., Xiong, H., Zheng, X.: A novel semantic matching method for indoor trajectory tracking. ISPRS Int. J. Geo-Inf. (2017)
15. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
16. Haddad, S., Wu, M., Wei, H., Lam, S.K.: Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. arXiv preprint arXiv:1902.05437 (2019)
17. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E (1995)
18. Hu, Y., Chen, S., Zhang, Y., Gu, X.: Collaborative motion prediction via neural motion message passing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
19. Huang, W., Han, J., Rong, Y., Xu, T., Sun, F., Huang, J.: Equivariant graph mechanics networks with constraints. In: International Conference on Learning Representations (2021)
20. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: STGAT: modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
21. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part IV 12, Springer (2012)
22. Kothari, P., Kreiss, S., Alahi, A.: Human trajectory forecasting in crowds: a deep learning perspective. IEEE Trans. Intell. Transp. Syst. (2021)
23. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer graphics forum. Wiley Online Library (2007)

24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
25. Mangalam, K., et al.: It is not the journey but the destination: endpoint conditioned trajectory prediction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer (2020)
26. Mantini, P., Shah, S.K.: Human trajectory forecasting in indoor environments using geometric context. In: Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (2014)
27. Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11. Springer (2010)
28. Rossi, L., Paolanti, M., Pierdicca, R., Frontoni, E.: Human trajectory prediction and generation using LSTM models and GANs. Pattern Recogn. (2021)
29. Rudenko, A., Kucner, T.P., Swaminathan, C.S., Chadalavada, R.T., Arras, K.O., Lilienthal, A.J.: Thör: Human-robot navigation data collection and accurate motion trajectories dataset. IEEE Robot. Autom. Lett. (2020)
30. Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O.: Human motion trajectory prediction: a survey. Int. J. Robot. Res. (2020)
31. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. Springer (2020)
32. Sampieri, A., et al.: Pose forecasting in industrial human-robot collaboration. In: European Conference on Computer Vision. Springer (2022)
33. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.: Learning to simulate complex physics with graph networks. In: International Conference on Machine Learning. PMLR (2020)
34. Satorras, V.G., Hoogeboom, E., Welling, M.: E (n) equivariant graph neural networks. In: International Conference on Machine Learning. PMLR (2021)
35. Skenderi, G., et al.: DOHMO: embedded computer vision in co-housing scenarios. In: 2021 Forum on specification & Design Languages (FDL). IEEE (2021)
36. Toaiari, A., et al.: Scene-pathy: capturing the visual selective attention of people towards scene elements. In: International Conference on Image Analysis and Processing. Springer (2023)
37. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. PMLR (2021)
38. Wang, P., Yang, J., Zhang, J.: Location prediction for indoor spaces based on trajectory similarity. In: 2021 4th International Conference on Data Science and Information Technology (2021)
39. Wang, P., Yang, J., Zhang, J.: Indoor trajectory prediction for shopping mall via sequential similarity. Information (2022)
40. Xu, C., Li, M., Ni, Z., Zhang, Y., Chen, S.: Groupnet: multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
41. Xu, C., Mao, W., Zhang, W., Chen, S.: Remember intentions: retrospective-memory-based trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)

42. Xu, C., et al.: Eqmotion: equivariant multi-agent motion prediction with invariant interaction reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
43. Yang, Y., Feng, Z., Song, M., Wang, X.: Factorizable graph convolutional networks. In: Advances in Neural Information Processing Systems (2020)
44. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16. Springer (2020)
45. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: agent-aware transformers for socio-temporal multi-agent forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
46. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: Sr-LSTM: state refinement for LSTM towards pedestrian trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)

Sensor-Agnostic Graph-Aware Kalman Filter for Multi-Modal Multi-Object Tracking

Depanshu Sani[1,2](✉), Anirudh Iyer[1], Prakhar Rai[1], Saket Anand[1],
Anuj Srivastava[2], and Kaushik Kalyanaraman[1]

[1] IIIT-Delhi, New Delhi, India
[2] Florida State University, Tallahassee, USA
`depanshus@iiitd.ac.in`
`https://sites.google.com/iiitd.ac.in/saga-kf`

Abstract. Recent progress in open-source object detection techniques has significantly advanced Multi-Object Tracking (MOT) methodologies, primarily under the tracking-by-detection paradigm. To enhance the robustness and reliability of MOT systems, recent research has proposed integrating information gathered from diverse sensors. However, many Kalman filter-based MOT approaches assume the independence of object trajectories, overlooking potential inter-object relationships. While some efforts have been made to incorporate these relationships, they often concentrate on learning feature representations to facilitate better association. Moreover, the existing filter-based method for estimating graphs from noisy data is unsuitable for *online* MOT applications. To alleviate these problems, we introduce a Sensor Agnostic Graph-Aware (SAGA) Kalman filter, which is the *first online* state estimation technique designed to *fuse multi-modal graphs* derived from noisy multi-sensor data. We validate the effectiveness of our proposed framework through extensive experiments conducted on both synthetic and real-world driving dataset (nuScenes). Our results showcase an improvement in MOTA and a reduction in estimated position errors (MOTP) and identity switches (IDS) for tracked objects using the SAGA-KF.

Keywords: Graph Kalman Filters · Graph Tracking · Multi-Object Tracking · Multi-Sensor Fusion

1 Introduction

MOT methods such as SORT [5], DeepSORT [24] and ByteTrack [28], rely on a Kalman filter-based probabilistic framework for estimating the state of *individual objects* over time. This model-based framework assumes all object trajectories

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_25.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 380–398, 2025.
https://doi.org/10.1007/978-3-031-78444-6_25

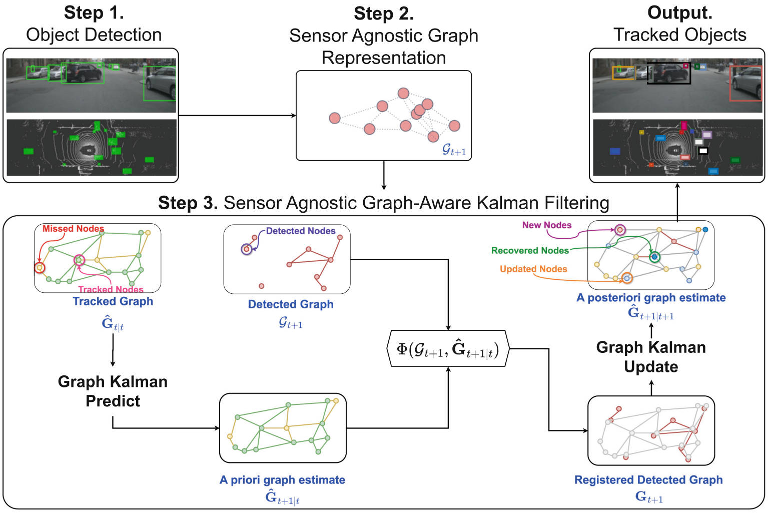


Fig. 1. We use state-of-the-art detectors to obtain objects in a scene and use them to construct the scene graph. The observed scene graph is then fed into the SAGA-KF framework to predict, associate and estimate the state of the scene graph. The proposed graph-based approach helps in better tracking by capturing the correlation between the dynamics of different objects. For instance, the observed neighbors of an occluded object can help better predict its position.

to be independent of each other and usually considers a constant velocity or acceleration model to predict the new state of each tracked instance. However, increasing scene complexity and the number of objects introduce additional challenges, such as detection errors (false alarms and misses) and occlusions. The nonlinear dynamics of individual objects also present significant challenges in achieving reliable tracking performance for safety-critical applications such as autonomous driving. This performance gap is often addressed by multi-modal sensor fusion using approaches like CLAMOT [26], TransFusion [2] and BEV-Fusion [16]. While several of these methodologies predict future positions by regressing velocities during the detection phase, certain approaches as Eager-MOT [12] adopt a formal model-based Kalman filter approach. While Kalman filters have been extensively used in single and multi-object tracking, the selected system evolutions (dynamical models) are often very basic. Most approaches track each object independently and in isolation from others, whereas, in application settings like road traffic monitoring, the dynamics of an object are arguably correlated with and are influenced by that of its neighbors. For instance, at a busy urban intersection, an autonomous vehicle can utilize the trajectory of an oncoming vehicle that is decelerating to anticipate the presence and movement of pedestrians who are temporarily occluded by a passing truck. This allows the autonomous vehicle to make safer and more informed navigation decisions, even when direct visibility is compromised.

In a recent paper, Bal et al. [3] seek to capture object dependencies using a graph-based representation that explicitly includes inter-object interactions. The objects of interest are represented as nodes of the graph, while interactions between the nodes are captured via edges. Using video as raw input, they formulate MOT as a *graph tracking* problem, which is solved by designing a Kalman filter over the *space of graphs*. Tracking a time-series of scene graphs, rather than individual nodes, helps in modeling the dependencies between the constituent objects. The parameters of the dynamical system are estimated using the classical maximum likelihood criterion. This approach was shown to improve estimation errors and better handle both false and missed detections while tracking multiple targets. However, tracking the entire scene graph results in a substantial increase in complexity; for an n-node graph with potentially n^2 edges, the covariance of the edge attributes requires estimating $\mathcal{O}(n^4)$ parameters. Moreover, the system parameters are estimated for each video, thereby limiting scalability. This, in turn, requires more hardware (memory and processing power), thus restricting its usage to video sequences that contain a small number of objects.

While [3] clearly establishes the benefits of capturing inter-object relationships using graph representations, its practical applicability and scalability is restricted due to its edge tracking, i.e., estimating mean and covariance of n^2 edge attributes. In this paper, we argue that the inter-object relationships can still be captured in the dynamical model without explicitly tracking graph edges. The resulting focus on node-only tracking can substantially reduce the computational complexity of the Kalman filter. We retain object interactions by imposing a more structured and topology-aware dynamical model on nodes. This model captures inter-node interactions and allows us to share dynamics across the interacting objects. As the edges are not explicitly tracked in this approach, our proposed approach is termed *graph-aware* Kalman filter, as opposed to a graph-tracking Kalman filter. We also emphasize that a graph-based approach allows each object in the scene to be represented as an abstract entity in the scene graph. This abstract representation of the dynamic scene using graph-based representation makes it viable to incorporate different sensing modalities by registering the abstract graphs obtained from each sensor using an assignment method, like the Hungarian algorithm. However, the measurement noise associated with each sensor and its corresponding pre-processing method needs to be modeled. Our dynamical model is also designed to handle the measurement noise associated with each sensor, thereby making the state evolution *sensor-agnostic*. With these two novel components, i.e., a topology-aware node-interacting dynamical model and sensor-agnostic state evolution, we propose a Sensor-Agnostic Graph-Aware Kalman Filter (SAGA-KF) and show its efficacy on the MOT problem using a synthetic dataset and the nuScenes [6] autonomous driving dataset. The overview of SAGA-KF is shown in Fig. 1 and following is a summary of our contributions:

1. We propose a novel dynamical model that captures the inter-object relationships in the form of a time-varying and topology-aware state-transition function on graph nodes. This function is parameterized by a small number of *fixed parameters* that are estimated using the training dataset.

2. We propose an observation model that is agnostic to the low-level processing of sensory data. This allows any pre-processing method to be used to construct nodes and edges in the graph. In our experiments, we use the camera and LiDAR modalities in conjunction with popular deep-learning-based 3D object detection methods for pre-processing the raw sensor data.
3. We adapt and apply the node-only Kalman Filter model from [3] and demonstrate that it scales easily to much larger graphs.
4. We evaluate our method on the nuScenes dataset [6] which is a large, real-world, autonomous driving dataset.

The rest of the paper is organized as follows. In Sect. 2, we discuss related work, followed by a description of our problem setup and an overview of [3] in Sect. 3. We then present our proposed model, SAGA-KF in Sect. 4. We present the experimental setup and the results and analysis in Sects. 5 and 6 respectively.

2 Related Works

There are several areas of vision research that are relevant in the current context. We discuss these works next.

Multi-Object Tracking: Recent MOT techniques broadly fall into two categories: (i) tracking-by-detection and (ii) simultaneous detection and tracking. The tracking-by-detection approach involves employing separate models for object detection and subsequent tracking. This methodology proves advantageous in scenarios where the objects are already localized, and the sole aim is to trace their trajectories. These detected objects may be supplied by an oracle or obtained from a state-of-the-art detector. Conversely, simultaneous detection and tracking methods, also known as *single-stage tracking*, are particularly valuable when there is no prior information about the objects present in a given scene. The primary focus of research, which is common to both paradigms, is developing better association methods. Notable association methods include model-based techniques such as Kalman filter (SORT [5], ByteTrack [28], AB3DMOT [22], Poly-MOT [14]), appearance-based methods like ReID embedding (Deep-SORT [24], JDE [21], FairMOT [29]) and motion prediction based approaches (CenterTrack [30], CenterPoint [25]). The majority of these works assume a predefined dynamical model for each node, *e.g.*, constant velocity or constant acceleration models. Thus, the trajectory of an object is assumed to be independent of the trajectory of other objects, which is rarely true in a real-world road setting.

Some of the recent works also propose variations to the standard Kalman filter pipeline used for MOT. BoT-SORT [1] introduces a Camera Motion Compensation module that solves the registration problem between frames at $t-1$ and t to return an affinity matrix, which provides the scale, rotation and translation of the frames. These transformations are then introduced into the Kalman filter update steps in order to account for the dynamic motion of the camera. In order to account for the dynamic and non-linear motion of the objects, OC-SORT [7] designs Kalman filter to be observation-centric instead of being estimation-centric. They highlight that even if an object can be associated again by SORT after a period of not being tracked, it will likely be lost again because its Kalman filter parameters may have already deviated substantially from the correct ones due to the temporal error magnification. To alleviate this problem, they propose Observation-centric Re-Update (ORU) to reduce the accumulated error. The key idea is to create dummy observations by generating a virtual trajectory by referring to the observations on the steps starting and ending the untracked period. They use these virtual trajectories to run the Kalman predict and update steps to account for the accumulated temporal error. While such methods modify the Kalman filter equations, the dynamical system still treats each object independently.

Graph-Based MOT: Recent works acknowledge the need to capture relationships between different objects. A natural way to accomplish this is to use graph representations. GNN3DMOT [23] is one of the earliest methods for online MOT using graph neural networks (GNNs). They identify that the key process for reducing confusion during the data association pipeline is to learn discriminative features for different objects. Hence, they introduce a GNN for feature interaction mechanism and also propose a novel joint feature extractor to learn appearance and motion features from 2D and 3D space simultaneously. Similarly, Liang et al. [15] propose to use graph convolution networks to exploit the cues from the *neighbor graph* of a target. GCNNMatch [19] and GNMOT [13] are a few other works that also use GNNs to improve the appearance and motion representations. Another line of work focuses on learning associations among different objects [8,11,19]. MotionTrack [20] uses a self-attention mechanism followed by a GCN-based architecture to learn the interactions between different tracklets and to predict the offsets from the previous frame that can be used with the detections for IoU-based association.

To our knowledge, [3] is the only work that develops a method for estimating graphs from noisy, cluttered, and incomplete data that is naturally extended to MOT. They introduce a quotient space representation of graphs that incorporates temporal registration of nodes (objects); then use that metric structure to impose a dynamical model on graph evolution. Finally, they derive a Kalman smoother, adapted to the quotient space geometry, to estimate dense, smooth trajectories of graphs. Although this approach has been demonstrated to reduce the rate of false and missed detections, it is practically limiting because of various factors. We will elaborate on these limitations in Sect. 3.2.

Sensor Fusion for MOT: Many practical systems employ multiple sensors with different imaging modalities for data acquisition. Different sensors can capture complementary information and fusing them provides a better understanding of the observed scene. For instance, (visible-light) cameras are better at capturing visual appearances but struggle with depth and 3D information of objects. On the other hand, LiDARs are better for 3D measurements but provide sparse and weak appearance information. Consequently, much effort has been invested in fusing information captured from multi-modal sensors. Most of the research in this domain for MOT is restricted to single-stage methods. A general approach to multi-sensor MOT is to extract features from each sensor separately (e.g., by using deep learning-based feature extractors) and aggregate all the extracted features. This unified feature representation is then adopted for various downstream tasks, such as MOT [2,16,26,27].

EagerMOT [12] is a simple two-stage tracking method that uses a model-based approach to integrate all available object observations from camera and LiDAR sensors to reach a well-informed interpretation of the scene dynamics. Their approach is to obtain detected objects from different sensors and fuse them into fused object instances, parameterized jointly in 3D and/or 2D space; followed by a two-stage association procedure. The first stage comprises of matching instances with 3D information (with/without 2D information) to existing tracks. In the second association stage, unmatched tracks from the previous timestep are matched with instances, localized only in 2D. The filtering approach adopted by EagerMOT is similar to CIWT [18] except that they maintain the 2D and 3D state of tracks independently; which is based on the constant velocity model.

3 Problem Setup

3.1 Preliminaries and Notation

Assume a dynamic scene that is observed by multiple sensors to generate a time-series of sensory data (e.g., a video sequence or a sequence of LiDAR scans). Without loss of generality, we can assume that the sensors are asynchronous, *i.e.*, at any time t the scene is observed by only one sensor. We use $\mathcal{T}$ to denote the length of the observation (or the number of frames), m_t for the total number of detected objects in a frame and n for the total number of instances (unique objects) observed. Let $\mathcal{G}_t(V_t, E_t)$ represent the scene graph observed at time t, in which the $m_t = |V_t|$ nodes are constructed using a pre-trained object detector $\psi^{(\mathtt{n})}(\cdot)$ and edges E_t are formed using the function $\psi^{(\mathtt{e})}(\cdot)$. While each edge instance in $\mathcal{G}_t$ captures the relationship between the corresponding node instances, we also define the set of *edge types*, $\mathcal{E}$, that are common across all scenes of a dataset. Examples of edge types for a traffic scene could be `car-car`, `pedestrian-pedestrian`, `car-traffic_light`, and so on. We shall use these edge types to develop our state transition function in our dynamical model.

Let $\Phi(\mathcal{G}_t, g)$ be the graph registration function that is used to match an observed graph $\mathcal{G}_t$ with another graph g, yielding the registered graph $\mathbf{G}_t$. Graph matching [9,17] is performed by identifying the *optimal* permutation of the nodes so as to obtain registration between nodes that correspond to the same object of interest. This is a difficult problem and is further exacerbated when the two graphs have a different number of nodes. In Sect. 4, we discuss our approach that builds upon the registration approach adopted in [3]. Consistent with the commonly used Kalman filter notation, $\hat{\mathbf{G}}_{t|t-1}$ denotes the registered graph using the *a priori* state estimate and $\hat{\mathbf{G}}_{t|t}$ denotes the registered graph using the *a posteriori* state estimate.

3.2 Kalman Filters for Video Graphs

In the Classical Kalman filter (C-KF), one uses state and observation models for each object to estimate their trajectory. For simultaneously tracking multiple objects, C-KF methods make use of domain knowledge for resolving the node associations across frames. C-KF methods typically assume that each object is independent of the other. Contrary to this assumption, Bal et al. [3] were the first to propose a model (BEVG-KF[1]) for using Kalman Filters directly on time series of graphs generated from video data [3]. A key idea introduced in [3] was to register the observed graph at time $t-1$, $\mathcal{G}_{t-1}$, with that at time t, i.e., $\mathcal{G}_t$, using a modified *Umeyama* algorithm developed for graph matching [10].

After the temporal graph registration, the Kalman filter is applied to the node and edge attributes. For temporally registered graphs, a linear discrete-time graph dynamical system as defined in [3] is given below, where $l \in \{node, edge\}$ represents the system equations for *node* or *egde* filtering respectively:

$$\begin{aligned}\mathbf{x}_t^{(l)} &= \mathbf{F}^{(l)}\mathbf{x}_{t-1}^{(l)} + \boldsymbol{\Omega}^{(l)}\mathbf{w}_{t-1}^{(l)}\\ \mathbf{y}_t^{(l)} &= \mathbf{W}_t^{(l)}\left(\mathbf{H}^{(l)}\mathbf{x}_t^{(l)} + \boldsymbol{\Lambda}^{(l)}\mathbf{v}_t^{(l)}\right)\end{aligned}$$

Here $\mathbf{x}_t^{(l)}$ is the state vector, $\mathbf{y}_t^{(l)}$ is the observation vector, $\mathbf{F}^{(l)}$ and $\mathbf{H}^{(l)}$ are the *time-invariant* state transition matrix and observation matrix respectively. The vectors $\mathbf{w}_t^{(l)}$ and $\mathbf{v}_t^{(l)}$ denote the random perturbations or additive noise from a standard normal distribution; the resulting covariance matrices for the process and observation noise are given by $\mathbf{Q}^{(l)} = \boldsymbol{\Omega}^{(l)T}\boldsymbol{\Omega}^{(l)}$ and $\mathbf{R}^{(l)} = \boldsymbol{\Lambda}^{(l)T}\boldsymbol{\Lambda}^{(l)}$, respectively. To deal with different number of nodes (object instances), m_t at different time steps, the knowledge of the maximum number of nodes (n) is assumed and $(n - m_t)$ *null nodes* are introduced at the t^{th} timestep. The matrix $\mathbf{W}_t^{(l)}$ in the observation equation above handles the difference between the number of nodes in the observed graph and the tracked graph. Further details of the Kalman equations as adapted by [3] are provided in Sect. A.1 of the supplementary material included with this paper. Additionally, the parameters of their dynamical model are estimated using a classical maximum likelihood based approach, also summarized in the supplementary material Sect. A.2. The model parameters are estimated for each video scene, implicitly assuming that the state-transition function is time-invariant, even when the scene graph topology changes over time. It is worth noting that the parameter estimation depends on the quality of graph registration, which is expected to be noisy due to the use of *observed* graphs. The registration method also makes use of all future observations, thus precluding the applicability to online tracking methods. Moreover, to deal with different number of nodes at each time step, null nodes are introduced, which further increase the number of model parameters to be estimated, the computational and the memory requirements. Finally, two independent dynamical models are used separately, one each for the nodes and the edges.

[1] Bayesian Estimation method for Video Graphs using Kalman Filters.

4 Graph-Aware Kalman Filter

To address the limitations discussed in the previous section, we propose a dynamical system that leverages the topological structure of the graph, without the need to track the complete graph. We take an *online* approach to tracking, i.e., processing incoming frames as they are streamed, as well as a sensor-agnostic approach, where the multi-modal observations are used to update a sensor-agnostic fused graph. With these additions, we optimize the implementation of the Kalman filter that is more memory and time efficient.

For defining a state estimation technique to be *online*, firstly, it is important to characterize the state of a node. Whenever a previously tracked object is not visible in any dynamic scene, it can either be occluded or moved out of the current frame. Now, it is crucial to decide whether such a node is expected to be returning to the scene in subsequent timesteps or not. If not, we don't want such nodes to contribute towards the estimation of the graph.

Definition 1. *For the tracked graph $\hat{\mathbf{G}}_{t|t}$, we categorize the state of a node as:*

- *Observed: Node is currently visible, i.e., detected by $\psi^{(n)}(\cdot)$ at time t.*
- *Missed: Node was visible at some time t', such that $(t - \delta t) \leq t' < t$.*
- *Dropped: Node is not visible since time $(t - \delta t)$.*

We propose to achieve a better graph matching by registering the observed graph $\mathcal{G}_t$ with the *a priori* estimate of the tracked graph $\hat{\mathbf{G}}_{t|t-1}$ (instead of the observed graph $\mathcal{G}_{t|t-1}$ as in BEVG-KF) using the Umeyama algorithm while also imposing the domain constraints, as in C-FK. Thus, a node in the observed graph that is not matched to any of the *observed* or *missed* nodes is considered a *new* node that needs to be inserted into the tracked graph. Similarly, whenever an *observed* or *missed* node is not matched to any of the nodes in the observed graph, the state of such nodes is updated based on Definition 1. Further, we assume that the *dropped nodes* are not expected to return; thus, such nodes only contribute towards the size and order of the graph while not affecting the state estimation. Therefore, removing the dropped nodes from the tracked graph will result in improving the space and time complexity. However, removing nodes from the tracked graph is non-trivial because the corresponding system matrices are associated with the entire graph and, therefore, will affect the Kalman equations, which assume a fixed size for system matrices. A similar argument holds whenever a *new* node needs to be inserted into the tracked graph. Later in this section, we redefine the Kalman equations for dynamic graphs with varying order and size. Since we now have a notion to actively add or drop the nodes, we first define the graph dynamical system while assuming that the system matrices only include the *observed* and *missed* nodes.

Definition 2. *We define the linear discrete-time graph-aware dynamical system as:*

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{w}_t \tag{1}$$

$$\mathbf{y}_t = \mathbf{W}_t(\mathbf{H}\mathbf{x}_t + \mathbf{v}_t) \tag{2}$$

where $\mathbf{F}_t$ is the time-varying state transition matrix; $\mathbf{W}_t$ is the matrix used to retain only the observed nodes that were tracked previously; all the other variables remain the same as before.

Note that, unlike BEVG-KF, the state and observation equations above do not include the process and observation covariances $\boldsymbol{\Omega}$ and $\boldsymbol{\Lambda}$; resulting in a reduction in the number of model parameters. Instead, we learn the process covariance $\mathbf{Q}$ and observation covariance $\mathbf{R}$ using the training dataset.

As discussed in the previous sections, a major drawback of BEVG-KF is that the dynamical system can only be defined for a single scene. Moreover, the number of model parameters grows substantially ($\mathcal{O}(n^4)$) with the number of instances (n) in the scene. To improve the generalizability of the dynamical system, it is crucial to reduce the number of parameters and define a state-transition function that is known a priori or can be reliably estimated from the state of the dynamical system itself. Furthermore, BEVG-KF maintains two separate and independent dynamical systems for nodes and edges. Therefore, the estimation of node attributes is not affected by the actual topology of the dynamic graphs defined via edges; rather, it learns a time-invariant state-transition function that might define a topology different from the one defined using the edges in the graph.

Definition 3. *We define the time-varying state-transition function as a linear combination of arbitrary but known state-transition functions $\mathbf{A}_t^e$ that encode the edge information of the tracked graph $\hat{\mathbf{G}}_{t|t}$. Specifically,*

$$\mathbf{F}_t = \tilde{\mathbf{F}}_t + \sum_{e \in \mathcal{E}} \mu_e \cdot \mathbf{A}_t^e \tag{3}$$

where $\tilde{\mathbf{F}}_t$ is the state transition function encoding the dynamics of each node independently (e.g., constant velocity model), $\mathcal{E}$ is the set of different edge types, $\mathbf{A}_t^e$ is an 'interaction function' which is a state-transition function based on the weighted adjacency matrix corresponding to edge type e and μ_e factors the impact of influence exerted by the edge e.

We point out that $\tilde{\mathbf{F}}_t$ is a block diagonal matrix with state-transition function corresponding to the independent motion dynamics of a node (e.g., a constant velocity model) as the block diagonal elements. Such a block diagonal matrix signifies that the state-transition of each node is not affected by any other node in the graph. Also, $\mathbf{A}_t^e$ can either be constructed based on the edges retrieved using a deep learning-based edge feature extractor, $\psi^{(\mathrm{e})}(\hat{\mathbf{G}}_{\mathbf{t}|\mathbf{t}})$, or can be handcrafted based on the current state estimate of the graph $\hat{\mathbf{G}}_{\mathbf{t}|\mathbf{t}}$. Thus, this formulation of state equations for graph spaces allows the model parameters $\boldsymbol{\mu}(= [\mu_1 \cdots \mu_{|\mathcal{E}|}]^T)$ to be independent of the order and size of the graph; and the number of parameters is significantly reduced to $|\mathcal{E}|$. Moreover, it supports online state estimation of graphs and the model parameters can be learned from a distinct set of scenes.

Let's illustrate the operational principle of the proposed state-transition function with a concrete example. For this example, let's assume that $\tilde{\mathbf{F}}_\mathbf{t}$ is a constant velocity model, $\mathcal{E} = \{\textit{`trailing'}\}$ and the edge weights are the scaled distances between the nodes. Let's define the interaction function so that the position of a node is updated based on the weighted average of the *rate of change in velocities* of the neighboring nodes, while all the other attributes are constant. Now, consider the following example.

"In an urban driving scenario, an autonomous vehicle in the rightmost lane may have its view of a smaller car (A) in the leftmost lane occluded by a large truck (B) in the middle lane. By observing the behavior of a leading vehicle (C) in the leftmost lane that starts to decelerate, the autonomous vehicle should be able to infer that the occluded smaller car will also decelerate."

Inferring this information from the dynamic scene is possible using SAGA-KF because the state evolution of A depends on its constant velocity and the interaction function, i.e., the *rate of change in velocity* of C, weighted based on the learned parameter $\mu_{\textit{`trailing'}}$. Therefore, when C starts to decelerate, the interaction function allows A to update its position by aggregating the *rate of change in velocities* of the neighboring nodes. This is the key idea when extended to graphs containing multiple neighbors and different edge types is effective for modeling more complex scenarios, e.g., a road traffic scene.

4.1 State Transition Model Parameters

With the goal of generalizability and viability for online state estimation, we propose a model such that its parameters can be learned from a separate set of *training* scenes. Let's assume $\mathbf{x}_t$ denotes the state vector obtained from ground truth labels, $\boldsymbol{\mu}$ denotes the model parameters, i.e., the column vector $(\mu_1, \mu_2, \cdots, \mu_{|\mathcal{E}|})$, and $\mathbf{A}_t$ is the matrix whose e'th column is the column vector $\mathbf{A}_t^e \mathbf{x}_t$. We then use least squares estimation to learn the model parameters $\boldsymbol{\mu}$. The final solution to estimate these parameters is given in Eq. (4). The steps to derive this solution are given in section B of the supplementary material.

$$\boldsymbol{\mu} = (\sum_{t=1}^{\mathcal{T}} \mathbf{A}_t^T \mathbf{A}_t)^{-1} (\sum_{t=1}^{\mathcal{T}} \mathbf{A}_t^T (\mathbf{x}_{t+1} - \tilde{\mathbf{F}}_t \mathbf{x}_t)) \tag{4}$$

Following an approach similar to BEVG-KF, we learn the covariance matrices from the data. The key difference in our approach is that we compute the covariance matrices for each node in the training dataset instead of a single scene. This helps us in constructing generic covariance matrices that can be used for any scene. We then stack the node-level covariance matrix to get a block diagonal covariance matrix representing the graph-level covariance matrix. A more detailed explanation about this approach is provided in section C of the supplementary material.

4.2 Observation Model Parameters

The observation noise covariance is estimated based on the detection errors, i.e., prediction errors between the detected objects and ground truth annotations.

This implies that the observation noise covariance is directly estimated using the deep-learning-based detector's predictions rather than the sensor itself. As a result, any pre-processing method can be used to construct the nodes and edges in the graph, making the approach *sensor-agnostic*. Consequently, our proposed observation model (Eq. 6) in the dynamical system is agnostic to the low-level processing of sensory data.

$$\mathbf{x}_t = \mathbf{F}_t\mathbf{x}_{t-1} + \mathbf{w}_t \tag{5}$$

$$\mathbf{y}_t^{(s)} = \mathbf{W}_t^{(s)}\left(\mathbf{H}\mathbf{x}_t + \mathbf{v}_t^{(s)}\right) \tag{6}$$

where (s) represents the different sensors involved. Equation (6) implies that the Kalman update step is sensor-dependent and that the observation noise matrix, $\mathbf{R}^{(s)}$, can be different for each sensors.

4.3 Kalman Filter Update Equations

We have defined the graph-aware dynamical system that handles graphs constructed from noisy, multi-sensor data having different order and size. The state transition function leverages the graph topology, and uses the training data to estimate the parameters for the state transition and the observation models. Next, we have to modify the Kalman equations proposed by BEVG-KF to account for the dynamic graphs with varying order and size. For this purpose, we introduce two operators $\widetilde{\mathbf{W}}_\mathbf{t}$ and $\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}$. Similar to the $\mathbf{W_t}^{(s)}$ matrix operator discussed above in Eq. (6), $\widetilde{\mathbf{W}}_\mathbf{t}$ retains *missed* nodes along with the previously tracked *observed* nodes and $\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}$ is used to insert new nodes. We assume that $\widetilde{\mathbf{W}}_\mathbf{t}$, $\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}$ and $\mathbf{W_t}^{(s)}$ have dimensions $(\tilde{w}_t \times n_t)$, $(\tilde{w}_t^{(i)} \times n_t)$ and $(w_t \times n_t)$ respectively, where n_t is the number of instances currently being tracked, $\tilde{w}_t \leq n_t$, $\tilde{w}_t^{(i)} \geq n_t$ and $w_t \leq n_t$. We adopt the same strategy as BEVG-KF to retain the relevant nodes, i.e., $\widetilde{\mathbf{W}}_\mathbf{t}$ can be constructed by removing the rows corresponding to the nodes to be ignored from an identity matrix $\mathbf{I}_{n_t}$. Similarly, $\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}$ can be constructed by appending a row of zeros to $\mathbf{I}_{n_t}$ for every node to be inserted. We give the algorithm for a single pass of the online state estimation of dynamic graphs with varying order and size below (also illustrated in Fig. 1). The shape of the resultant matrix, wherever needed, is mentioned with blue color.

1. Construction of a scene graph from noisy sensor data observation, $\mathcal{G}_t$.
2. Kalman predict (*a priori* estimation of the tracked graph to obtain $\hat{\mathbf{G}}_{t|t-1}$):
   (a) A *priori* state estimation $(\tilde{w}_t \times 1)$:
   $\hat{\mathbf{x}}_{t|t-1} = (\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{F}_t\widetilde{\mathbf{W}}_\mathbf{t}^T)(\widetilde{\mathbf{W}}_\mathbf{t}\hat{\mathbf{x}}_{\mathbf{t-1|t-1}})$
   (b) A *priori* covariance estimation $(\tilde{w}_t \times \tilde{w}_t)$:
   $\mathbf{P}_{t|t-1} = (\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{F}_t\widetilde{\mathbf{W}}_\mathbf{t}^T)(\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{P}_{t-1|t-1}\widetilde{\mathbf{W}}_\mathbf{t}^T)(\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{F}_t^T\widetilde{\mathbf{W}}_\mathbf{t}^T) + (\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{Q}\widetilde{\mathbf{W}}_\mathbf{t}^T)$
3. Registration of observed graph $\mathcal{G}_t$ with *a priori* graph estimate $\hat{\mathbf{G}}_{t|t-1}$ using the graph registration module $\Phi(\mathcal{G}_t, \hat{\mathbf{G}}_{t|t-1})$.
4. Kalman update (*a posteriori* estimation of the tracked graph to obtain $\hat{\mathbf{G}}_{t|t}$):

(a) Kalman Gain $(\tilde{w}_t \times w_t)$:

$$\mathbf{K}_t^{(s)} = \mathbf{P}_{t|t-1}(\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{H}^T\widetilde{\mathbf{W}}_\mathbf{t}^T)(\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{W}_t^{(s)T})(\mathbf{W}_t^{(s)}\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T\mathbf{W}_t^{(s)T} + \mathbf{W}_t^{(s)}\mathbf{R}^{(s)}\mathbf{W}_t^{(s)T})^{-1}$$

(b) A *posteriori* state estimation $(\tilde{w}_t \times 1)$:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t^{(s)}(\mathbf{W}_t^{(s)}\mathbf{y}_t^{(s)} - \mathbf{W}_t^{(s)}\mathbf{H}\hat{\mathbf{x}}_{t|t-1})$$

(c) A *posteriori* covariance estimation $(\tilde{w}_t \times \tilde{w}_t)$:

$$\mathbf{P}_{t|t} = (\widetilde{\mathbf{W}}_\mathbf{t}\mathbf{I}\widetilde{\mathbf{W}}_\mathbf{t}^T - \mathbf{K}_t^{(s)}\mathbf{W}_t^{(s)}\mathbf{H}\widetilde{\mathbf{W}}_\mathbf{t}^T)\mathbf{P}_{t|t-1}$$

5. Insertion of new nodes, i.e., observed nodes not previously tracked:

(a) Inserting rows to the state vector $(\tilde{w}_t^{(i)} \times 1)$:

$$\hat{\mathbf{x}}_{t|t} = \widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}\hat{\mathbf{x}}_{t|t}$$

(b) Inserting rows and columns to the state-transition $(\mathbf{F}_\mathbf{t})$, observation $(\mathbf{H})$, process covariance $(\mathbf{P}_{t|t})$ and noise matrices ($\mathbf{Q}$ and $\mathbf{R}^{(s)}$). Let's denote all these matrices using $\mathbf{X}$ $(\tilde{w}_t^{(i)} \times \tilde{w}_t^{(i)})$:

$$\mathbf{X} = \widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}\mathbf{X}\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})T}$$

(c) Initializing the new rows and columns as discussed in the next section.

4.4 Optimization of the Graph Filtering Method

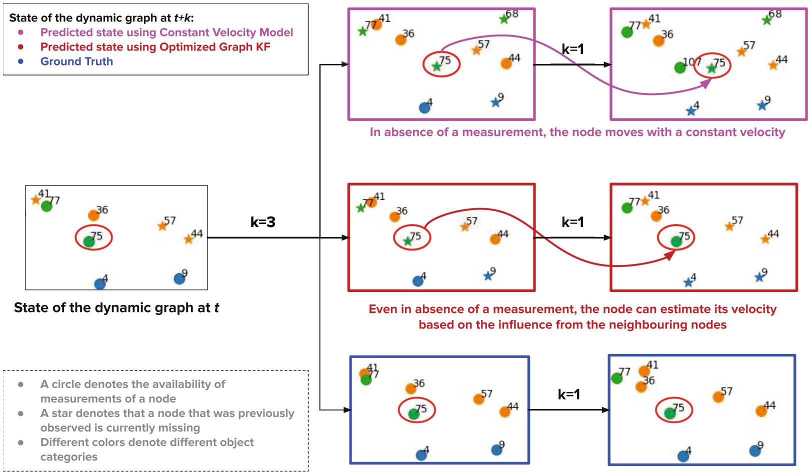
By defining the notion of actively adding or deleting the nodes, we modified the Kalman equations for dynamic graphs with varying sizes and order. As discussed earlier, this is facilitated by introducing two new operators $\widetilde{\mathbf{W}}_\mathbf{t}$ and $\widetilde{\mathbf{W}}_\mathbf{t}^{(\mathbf{i})}$. Although employing these operators reduces the size of the state vector and process covariance matrix significantly, the number of matrix multiplications is increased considerably. Thus, we can physically remove the corresponding rows and columns from the system matrices to improve the space and time complexity. This will significantly reduce the size of all the system matrices and the number of matrix multiplications. It is important to note that employing this optimization reduces the Kalman equations to the one proposed by BEVG-KF with an added assumption that size of the state and system matrices is variable and they only include the *observed* and *missed* nodes. This optimization enables scalability, which is critical for applications like Multi-Object Tracking. A long driving sequence can easily have 1000 unique object instances, which will lead to a 1000-node graph. While BEVG-KF will need to estimate 10^6 parameters for node-tracking and 10^{12} parameters for edge-tracking, this optimization limits the size of our graph to the maximum number of unique objects in a short window (δt) of the driving sequence.

5 Experimental Setup

5.1 Experimental Configuration

Baselines: We use C-KF, a node-level model-based tracking approach, and CenterPoint [25], a velocity regression-based tracking approach, as the baseline methods. Most of the recent tracking-by-detection methods employ a state-of-the-art detector along with these tracking approaches. Due to the limitations of BEVG-KF [3] highlighted in the previous sections, we do not compare it with our approach. However, we compare the computational expense incurred while using [3] as compared to SAGA-KF, as discussed in the next section.

Graph Filtering Optimization: We show the benefits of optimization on the mini-val set of nuScenes dataset (i.e. 2 scenes). We use a graph representation of fixed order and size, as in BEVG-KF, and compare the time taken to track a graph as the maximum number of instances increases. For a fair comparison, we assume that the graph has no edges and use the constant-velocity model for all the methods, thus yielding the same MOT performance. Whenever a new node was observed, which can not be tracked because of the limit on the number of instances, we performed a regression-based prediction and update step as in [25].


State of the dynamic graph at *t+k*:
Predicted state using Constant Velocity Model
Predicted state using Optimized Graph KF
Ground Truth
k=3
k=1
k=1
k=1
41 77 36 57 75 44 4 9
77 41 68 36 57 75 44 4 9
77 41 68 36 57 107 75 44 4 9
77 41 36 57 75 44 4 9
41 77 36 57 75 44 4 9
State of the dynamic graph at *t*

In absence of a measurement, the node moves with a constant velocity

Even in absence of a measurement, the node can estimate its velocity based on the influence from the neighbouring nodes

- A circle denotes the availability of measurements of a node
- A star denotes that a node that was previously observed is currently missing
- Different colors denote different object categories

Fig. 2. A snapshot from the synthetic dataset. This figure demonstrates the influence of neighboring nodes on the dynamics of a node.

Graph Registration Module: The graph registration module, $\Phi(\cdot, \cdot)$, takes two scene graphs as inputs. For each node in the observed graph $\mathcal{G}_t$, we identify a subset of nodes from $\hat{\mathbf{G}}_{t|t-1}$ such that the Euclidean distance between the observed node and the tracked node is less than a certain threshold. To compute the cost of matching the entire graphs, we calculate the Mahalanobis distance between the observed node's attributes and the process covariance of the tracked node in its identified subset. As no covariance is associated with the regression-based approach, we use the Euclidean distance instead of Mahalanobis for CenterPoint. Like BEVG-KF, we use the Umeyama matching algorithm for SAGA-KF and the Hungarian matching algorithm for C-KF and CenterPoint.

Datasets: We use a synthetic dataset to present our results followed by comparative evaluation on nuScenes [6], a large, real-world, autonomous driving dataset.

A snapshot of the synthetic dataset is shown in Fig. 2 and the details of both, the synthetic and nuScenes dataset along our scene graph construction approach are also provided in the material. To evaluate the effectiveness of the proposed SAGA-KF, we evaluate its performance for a single and multi-sensor setup. To simulate the effect of false negative detection errors, in the case of synthetic data, we randomly drop nodes in the scene graph at each time. We then compare the MOT metrics obtained using our approach with the baselines when different levels of detection errors are introduced.

5.2 MOT Metrics

We use the CLEAR-MOT [4] metrics to evaluate the performance of the synthetic data. We highlight the performance improvements achieved in MOT Accuracy (MOTA), MOT Precision (MOTP) and the number of Identity Switches (IDS). For the nuScenes dataset, we use their official evaluation pipeline and additionally report the Average MOTA (AMOTA) and Average MOTP (AMOTP).

6 Results and Analysis

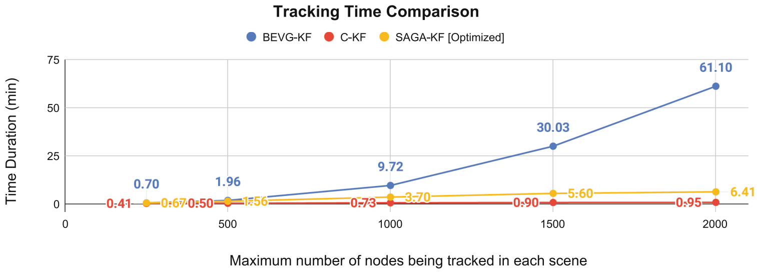


Fig. 3. Tracking time comparison using the corresponding Kalman equations.

We show the efficiency gains attained through the optimization in Fig. 3. It can be observed that the execution time is the least in the case of C-KF because the size of the state vector and system matrices is independent of the number of instances. The execution time increases drastically as the order of the graph increases in the case of BEVG-KF. This is primarily because the size of the state vectors and system matrices increases by $\mathcal{O}(nf)$ and $\mathcal{O}(n^2f^2)$, respectively, whenever the number of instances is increased by n and the number of attributes for each node is f. With the optimized graph filtering approach, the size is dependent only on the number of *active* nodes at any particular time, i.e., the number of *observed* and *missed* nodes. For an evolving scene, such as nuScenes, the number of *active* nodes is significantly less than the total number of instances. Hence, a significant reduction in execution time can be observed with the optimized BEVG-KF method. It is important to note that this experiment highlights the performance when only the nodes are tracked. This implies that the execution time for BEVG-KF would be substantially higher as compared to SAGA-KF when edge tracking is also included.

The summary of the results obtained for the multi-sensor synthetic dataset is shown in Table 1. The table clearly shows the benefits of using a graph-filtering approach as compared to C-KF for single sensor as well as fusion of multi-modal sensors. The reported metrics show a gradual increase in the error rate as the number of randomly dropped nodes increases. We also demonstrate the benefits of using the proposed sensor-agnostic graph fusion technique followed by the proposed graph Kalman filter. Significant improvement is observed throughout all the experiments with respect to all the reported MOT metrics. Figure 2 shows a snapshot from the synthetic dataset. This figure illustrates the qualitative performance and benefits of SAGA-KF as compared to C-KF.

Table 1. Ablative study on the synthetic dataset. 'Dropped Nodes' represents the % of the instances that were randomly dropped to simulate the effect of detection errors. A and B denote the two sensors, and A+B denotes their fusion.

Dropped	Tracking	MOTA (↑)			MOTP (↓)			IDS (↓)		
Nodes	Method	A	B	A + B	A	B	A + B	A	B	A + B
0%	Centerpoint	97.200	97.847	97.227	**0.569**	**0.665**	0.586	160	130	153
	C-KF	97.867	96.227	94.933	0.979	1.921	0.864	114	187	171
	SAGA-KF	**99.593**	**99.680**	**98.287**	0.767	1.162	**0.512**	**24**	**14**	**34**
20%	Centerpoint	93.940	94.780	92.193	1.248	**1.278**	0.997	424	363	422
	C-KF	95.127	93.907	91.700	1.394	2.237	1.278	335	404	343
	SAGA-KF	**95.973**	**96.327**	**94.493**	**1.231**	1.582	**0.864**	**276**	**246**	**193**
40%	Centerpoint	79.400	80.147	88.887	1.745	**1.716**	1.435	1223	1170	697
	C-KF	**80.627**	79.553	89.173	1.765	2.431	1.675	**1144**	1197	603
	SAGA-KF	80.133	**80.467**	**89.773**	**1.682**	2.002	**1.353**	1155	**1111**	**524**
60%	Centerpoint	48.600	49.207	77.073	1.99	**1.905**	1.913	2261	2219	1514
	C-KF	48.853	48.827	77.787	1.963	2.416	2.068	**2236**	2231	1431
	SAGA-KF	**48.887**	**49.607**	**77.900**	**1.905**	2.148	**1.857**	2237	**2195**	**1389**
80%	Centerpoint	**15.153**	**15.467**	37.273	**1.683**	**1.615**	**2.017**	**2142**	**2128**	2751
	C-KF	15.093	15.153	37.320	1.717	1.923	2.116	2143	2134	2738
	SAGA-KF	15.133	15.267	**37.507**	1.684	1.809	2.037	2143	2133	**2722**

For the real-world dataset, we observe improvements over the baseline methods, though the gains are marginal (Table 2). Therefore, we conduct ***additional experiments*** to better understand the advantages and limitations of our proposed technique. For this purpose, we first summed up the estimation errors (ϵ_s) of the predicted object trajectories for each scene. Then, we compared the difference between the summed errors obtained using C-KF ($\sum \epsilon_s^C$) and SAGA-KF ($\sum \epsilon_s^S$). We observed that out of the 150 validation scenes, SAGA-KF performs better on 69 scenes only, i.e., $\sum \epsilon_s^S < \sum \epsilon_s^C$. For 76 of the remaining validation scenes, our approach performs nearly the same as C-KF, i.e., $\sum \epsilon_s^S - \sum \epsilon_s^C \leq 0.5$ meters. This analysis points out the edges' and interaction functions' lack of ability to generalize well on all the scenes. To validate this hypothesis, we conducted an experiment wherein we assumed that the observations have no registration or observation errors by using the ground truth annotations. Again, we observed that SAGA-KF performs marginally better than C-KF. Since we know that whenever there are no observation or registration errors, the estimation errors using the proposed SAGA-KF depend on the weighted adjacency matrix and interaction functions. Therefore, we speculate that the potential of formulating the tracking problem as the SAGA-KF is evident from the improvement obtained in constrained and real-world environments. However, its performance depends on the reliability of the graph and interaction functions to estimate the real-world correlations. Additionally, we hypothesize that a better choice of edge connections and interaction functions can further improve the results.

Table 2. Results for the nuScenes dataset on the validation set.

Sensor	Tracking Method	AMOTA (↑)	AMOTP (↓)	MOTA (↑)	MOTP (↓)	IDS (↓)
Camera	CenterPoint	**36.84%**	**1.0101**	**39.03%**	**0.6780**	870
	C-KF	33.57%	1.0374	37.87%	0.7329	**398**
	SAGA-KF	33.57%	1.0375	37.95%	0.7344	399
LiDAR	CenterPoint	53.05%	**0.7062**	44.56%	0.3609	2521
	C-KF	53.77%	0.7422	45.80%	0.4459	**1363**
	SAGA-KF	**53.82%**	0.7418	**45.85%**	**0.4458**	1392
Fusion	CenterPoint	34.99%	0.7626	33.78%	**0.3913**	2099
	C-KF	39.71%	0.6969	**38.97%**	0.4557	960
	SAGA-KF	**39.76%**	**0.6950**	38.86%	0.4555	**946**

7 Discussion

The experiments with the synthetic dataset show that SAGA-KF clearly outperforms the other MOT methods. For the synthetic dataset, the graph construction was perfect, i.e., the method to construct edges used for describing the correlation of nodes in the dynamic scene was the same for data generation and observation. Moreover, the interaction functions were assumed to be known a priori; consequently, we had good estimates of the model parameters $\boldsymbol{\mu}$. This confirms that the correct graph construction mechanism and interaction functions can allow SAGA-KF to better estimate the scene evolution by incorporating the relationships between nodes without explicitly tracking the edges. As observed in the previous section, the small performance gains in the real dataset may be due to sub-optimal graph construction and the handcrafted interaction function design. These results suggest that a more accurate interaction function, perhaps learned via data-driven methods can further benefit the model. While we used only camera and LiDAR in this paper, the approach is agnostic to the sensor and pre-processing method and can be used for other sensing modalities like depth or thermal cameras or RADARs with associated processing techniques.

8 Conclusion

This work develops a Kalman filter that uses a graph-based state representation. The graph-based representation allows different objects in the scene to interact, thus capturing their correlation while estimating the state for MOT. Therefore, estimating the state of the graph implicitly estimates the state of each object in the scene while capturing how the other neighboring objects influence the dynamics of the object. The proposed solution, SAGA-KF, develops a novel MOT technique that efficiently encodes the edge information as interaction functions, which are used for tracking the nodes in the graph. SAGA-KF is also designed to maintain a common state for the entire scene while including observations from multiple sensors. By encoding the edge information as interaction functions into node-tracking, the proposed solution eliminates the need to track the edges, thereby, significantly reducing complexity. These interaction functions define a structure on the evolving scene by describing how the dynamics of any node are influenced by the node attributes of its neighbors. Moreover, by including observations from multiple asynchronous sensors into the common graph-based state representation, this method allows seamless integration of heterogeneous data from multi-modal sensors for MOT.

Acknowledgement. This work was supported by the NSF-TIH grant <TIH/iHub Drishti/Project/2022-23/22>. We are grateful for the support from the TIH, iHub Drishti at IIT Jodhpur and the Infosys Center for Artificial Intelligence at IIIT-Delhi. We also appreciate the contributions of Aditi Basu Bal and Nilay Shrivastava during the early stages of the project.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-Sort: Robust Associations Multi-pedestrian Tracking (2022). arXiv preprint: arXiv:2206.14651
2. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: TransFusion: Robust Lidar-Camera Fusion for 3D Object Detection with Transformers. CVPR (2022)
3. 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)
4. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP J. Image Video Process. **2008**, (2008)
5. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468 (2016). 10.1109/ICIP.2016.7533003

6. Caesar, H., et al.: nuscenes: a multimodal dataset for autonomous driving. In: CVPR (2020)
7. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9686–9696 (2023)
8. Ding, S., Rehder, E., Schneider, L., Cordts, M., Gall, J.: 3dmotformer: graph transformer for online 3d multi-object tracking. In: ICCV, pp. 9784–9794 (2023)
9. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE TPAMI **18**(4), 377–388 (1996)
10. Guo, X., Srivastava, A., Sarkar, S.: A quotient space formulation for generative statistical analysis of graphical data. J. Math. Imaging Vis. **63**(6), 735–752 (2021)
11. Jiang, X., Li, P., Li, Y., Zhen, X.: Graph Neural Based End-to-End Data Association Framework for Online Multiple-Object Tracking (2019). ArXiv: **abs/1907.05315**
12. Kim, A., Ošep, A., Leal-Taix'e, L.: Eagermot: 3D multi-object tracking via sensor fusion. In: IEEE ICRA (2021)
13. Li, J., Gao, X., Jiang, T.: Graph networks for multiple object tracking. In: Proceedings of the IEEE WACV (2020)
14. Li, X., et al.: Poly-mot: a Polyhedral Framework for 3D Multi-object Tracking, pp. 9391–9398. In: IROS (2023)
15. Liang, T., Lan, L., Luo, Z.: Enhancing the association in multi-object tracking via neighbor graph. Int. J. Intell. Syst. **36**, 6713–6730 (2020)
16. Liu, Z., et al.: Bevfusion: multi-task multi-sensor fusion with unified bird's-eye view representation. In: IEEE ICRA (2023)
17. Lyzinski, V., Fishkind, D.E., Fiori, M., Vogelstein, J.T., Priebe, C.E., Sapiro, G.: Graph matching: relax at your own risk. IEEE TPAMI **38**(01), 60–73 (2016)
18. Ošep, A., Mehner, W., Mathias, M., Leibe, B.: Combined image- and world-space tracking in traffic scenes. In: ICRA (2017)
19. Papakis, I., Sarkar, A., Karpatne, A.: A graph convolutional neural network based approach for traffic monitoring using augmented detections with optical flow. In: IEEE ITSC, pp. 2980–2986 (2021)
20. Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., Tang, W.: Motiontrack: learning robust short-term and long-term motions for multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17939–17948 (2023)
21. Wang, Z., Zheng, L., Liu, Y., Wang, S.: Towards real-time multi-object tracking. In: The European Conference on Computer Vision (ECCV) (2020)
22. Weng, X., Wang, J., Held, D., Kitani, K.: AB3DMOT: a Baseline for 3D Multi-Object Tracking and New Evaluation Metrics. ECCVW (2020)
23. Weng, X., Wang, Y., Man, Y., Kitani, K.: GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with 2D-3D Multi-Feature Learning. CVPR (2020)
24. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: IEEE ICIP, pp. 3645–3649 (2017)
25. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. CVPR (2021)
26. Zhang, S., Liu, X., Tao, W.: Clamot: 3d detection and tracking via multi-modal feature aggregation. In: Proceedings of the 4th International Conference on Image Processing and Machine Vision, pp. 27–31. ACM (2022)
27. Zhang, W., Zhou, H., Sun, S., Wang, Z., Shi, J., Loy, C.: Robust multi-modality multi-object tracking. In: IEEE ICCV (2019)

28. Zhang, Y., et al.: Bytetrack: multi-object tracking by associating every detection box. In: ECCV (2022)
29. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: on the fairness of detection and re-identification in multiple object tracking. IJCV **129**, 3069–3087 (2021)
30. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. ECCV (2020)

Multi-Level Feature Exploration Using LSTM-Based Variational Autoencoder Network for Fall Detection

Anitha Rani Inturi[1], V. M. Manikandan[1](✉), Partha Pratim Roy[2], and Byung-Gyu Kim[3]

[1] SRM University-AP, Neerukonda 522240, India
{anitharani.i,manikandan.v}@srmap.edu.in
[2] IIT Roorkee, Roorkee, Uttarakhand, India
partha@cs.iitr.ac.in
[3] Division of Artificial Intelligence Engineering, Sookmyung Women's University, Seoul 04310, Republic of Korea
bg.kim@sookmyung.ac.kr

Abstract. Accidental falls and their consequences are critical concerns for elderly people. Fatal injuries, when delayed in treatment, can lead to severe outcomes. Fall detection systems are crucial for the timely treatment of such injuries. Although sensor-based fall detection approaches are effective, video-based approaches are more useful because they assist in analyzing the fall scene and identifying the cause of the fall. However, privacy preservation is a major concern in video-based fall detection. The proposed system introduces a privacy-preserving mechanism that masks the identified human with a silhouette. A custom dataset, including 80 activities of daily living and 70 fall activities, is introduced. An LSTM variational autoencoder architecture is designed with a gradient clipping mechanism and a smooth variant of Adaptive Moment Estimation with Stochastic Gradient Descent (AMSGrad) optimizer to enhance the accuracy of fall detection. The reconstruction error between normal and fall activities is clearly identified with the help of a dynamic threshold. This results in a system performance that achieves accuracy, precision, and sensitivity of 99%, 97%, and 99%, respectively.

Keywords: Fall Detection · Computer Vision · Assistive Living · Autoencoders · Deep learning

1 Introduction

In the contemporary world, life expectancy has increased due to substantial advancements in medicine and technology. According to the World Health Organization (WHO), one in six individuals worldwide will be 60 years of age or older by 2030. This will result in an increase in the population of people over 60 from 1 billion in 2020 to 1.4 billion in 2030 [24]. Therefore, it is imperative to address

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 399–414, 2025.
https://doi.org/10.1007/978-3-031-78444-6_26

the primary issues encountered by the geriatric population. Falls are a major concern for the elderly and people with chronic illnesses. An incident where a person unintentionally descends to a lower level or the ground is considered a fall. However, events caused by acute illness or environmental hazards are not classified as falls. The number of deaths caused by unintentional falls in the geriatric population is increasing rapidly every year. The Centers for Disease Control and Prevention (CDC) conducted a thorough analysis investigating the number of deaths related to inadvertent falls between 2018 and 2023, as shown in Fig. 1 [9]. According to a review of 19 studies, the percentage of falls among older Indian adults ranged from 14% to 53%. This situation has led to increased research efforts focused on creating and exploring assistive living environments.

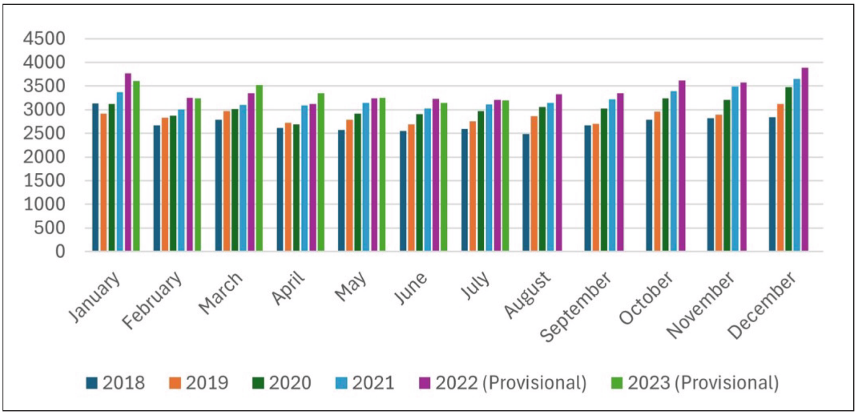


Fig. 1. A study given by the Centers for Disease Control and Prevention on deaths caused by unintentional falls in the geriatric population from the year 2018 to 2023

Aging is the main cause of falls among the elderly. An array of factors, including urbanization, globalization, and technological advancements, directly or indirectly impact the elderly population. The years 2021 to 2030 have been designated by the UN as the Decade of Healthy Aging to address this issue. In this manner, suitable solutions to address the various factors affecting senior citizens can be developed. Researchers attempting to address falls are working to develop reliable fall detection systems.

Due to the high sensitivity required in fall detection systems, significant attention is devoted to research in this area. Fall detection techniques are broadly classified into ambient sensors, vision-based techniques, and wearable sensors. This research primarily focuses on vision-based techniques. The advantages of vision-based approaches are comparatively higher than other approaches. Research has shown that ambient sensors and wearable sensors do not provide any information additional to their vision-based counterparts [19].

Vision-based methods are further divided into RGB and depth visual categories based on the type of camera used. A combination of global and local features is used to analyze the RGB images. Global features are characteristics of the image as a whole or of a particular area, while local features are characteristics of a specific area or segment, such as color, texture, intensity, or distinct elements. In contrast, depth features in depth images are non-intrusive and robust to variations in illumination. These characteristics provide depth values that enhance the analysis of vision-based systems. Each category has its own set of benefits and drawbacks. Depth-based methods are unsuitable for contextual analysis since background information is lost, and the main drawback of RGB images is their lack of privacy [21].

The proposed approach uses the RGB visual category for the development of the fall detection framework. The framework integrates a privacy-preserving mechanism to mitigate the drawbacks of RGB images. An imbalance in data results from the significant variance in the number of daily activities that the elderly population engages in compared to the yearly incidence of falls. Algorithms that rely on balanced data cannot effectively address the fall detection problem due to this identified deviation. The evolution of machine learning and deep learning architectures has addressed the problem of dealing with unbalanced data.

This paper proposes a fall detection framework where falls are considered anomalies while other activities are considered normal. A variational autoencoder is designed using long-short term memory (LSTM) layers. This LSTM variational autoencoder (LSTM-VAE) is trained on normal activities and tested on both fall and normal activities. The LSTM encoder is used to examine the temporal sequences, which are further projected onto a lower-dimensional latent space. Later, the LSTM decoder reconstructs the sequence. Any sequence that deviates from the set sequence is considered an anomaly. The main contributions of the proposed work are:

- A dataset has been created with 30 falling activities and 40 daily living activities.
- A privacy-preserving technique is introduced, ensuring that the people being captured by the camera are concealed during fall analysis.
- An LSTM-VAE is designed and optimized by implementing gradient clipping and a smoothened AMSGrad optimizer.
- Since 80 daily living activities are considered for training the system, the number of false alarms has been reduced.

This research article is organized as follows: Related work is discussed in Sect. 2, the proposed work is detailed in Sect. 3, and the results obtained are presented in Sect. 4 along with a comparison with state-of-the-art algorithms in the literature. The conclusions and further improvements are discussed in Sect. 5.

2 Related Work

Machine learning algorithms apply optimization techniques such as gradient descent [15] to reduce the loss function. These techniques iteratively adjust the model parameters toward the lowest value of the loss function. In deep learning architectures, a backpropagation algorithm is employed to update the network parameters using optimization algorithms such as gradient descent. By updating the model parameters in the opposite direction of the gradients, the loss is minimized. Equation (1) is used recurrently to complete this update.

$$\theta_{(t+1)} = \theta_t - \alpha * \nabla L(\theta_t) \tag{1}$$

Though implementing gradient descent with backpropagation is simple and computation is straightforward, the model parameters are updated after the forward pass is completed on the whole dataset. This results in misleading local minima [22]. Also, the convergence of gradient descent is very slow when the difference in the computed gradients is very low. Alternatively, the stochastic gradient descent (SGD) method is capable of reaching global optimality despite misleading local minima [6,11,22]. This is achieved by updating model parameters on a random subset of the dataset using Eq. (2).

$$\theta_{(t+1)} = \theta_t - \eta * \nabla L(\theta_t; x^i, y^i) \tag{2}$$

where x^i, y^i are the input and target of the i^{th} training sample. However, this update results in high variance among the parameters and an unstable gradient descent. Alternatively, the mini-batch SGD computes gradients over small batches of data using Eq. (3). This reduces the variance among the parameters and supports a stable gradient descent.

$$\theta_{(t+1)} = \theta_t - \eta * \nabla L(\theta_t; x^i, y^i) \tag{3}$$

where x^i, y^i are mini-batches of input and target.

2.1 Adaptive Learning Rates

Adaptive learning rates can be classified based on their application to stationary or non-stationary objectives. These techniques learn from the gradient information and adjust the learning rate of the training process dynamically. While learning rates for stationary objectives do not retain all sequences of information, those for non-stationary objectives retain important information and avoid the vanishing gradient concern. The work described in [3] evaluated their proposed model using various adaptive learning rate algorithms such as adaptive moment estimation (ADAM), adaptive gradient algorithm (ADAGrad), root mean square propagation (RMSProp), etc. These algorithms typically use an approach to filter out old data, either by continuously monitoring modifications or keeping an up-to-date estimate. As a result, their predictions are affected by the vanishing gradient problem. Learning rates for non-stationary objectives observe unknown changes in the parameters, also known as concept drift. These techniques are very useful in time series analysis. Adaptive moment estimation with stochastic gradient descent (AMSGrad) [7] is one such optimization algorithm that supports time series analysis. A maximum of the squared gradients is used to update the parameters of the optimization function as given in Eq. (4).

$$\theta_{(t+1)} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} * m_t \tag{4}$$

where m_t and v_t represent the first and second gradient moments. While the first moments estimate the direction of gradients, the second moments analyze the variance of gradients across iterations. The work proposed in this paper customizes an AMSGrad gradient descent algorithm to overcome the convergence problem when it gets stuck in local minima.

2.2 Fall Datasets

- SDUFall Dataset: The authors in [13] developed this action dataset using low-cost Kinect depth cameras. The activities consisted of 5 activities of daily living (ADL) and 1 fall activity.
- CMD-FALL Dataset: This dataset comprises multimodal multiview data that includes RGB images, depth images, and sensor data. The authors in [20] captured activities from 50 subjects, each performing 8 falling activities and 12 non-fall activities.
- Fall Dataset: The authors of the fall dataset [1] recorded falls in different indoor setups. The dataset is a collection of RGB images, depth images, and their combination RGB-D images.
- UP-Fall Dataset: This is a multimodal dataset developed by [14]. The dataset comprises 11 activities, out of which 6 are ADL and 5 are falling activities performed by 17 subjects.
- URFall Dataset: The authors of this dataset [10] employed two Kinect cameras, one parallel to the ceiling and the other parallel to the floor. The dataset consists of 40 activities of daily living and 30 fall activities.

It is observed that the amount of data available for research on identifying falls is very limited. Hence, the proposed work constructs a dataset that covers 80 ADL and 70 fall activities.

2.3 Fall Detection Approaches

Recent advancements in fall detection have significantly improved the accuracy and robustness of various methods. The work [2] presents an automated vision-based fall detection system that achieves 100% accuracy using SVM classifiers on three benchmark datasets. The system uses human segmentation, image fusion, and a 4-stream 3D convolutional neural network (4S-3DCNN) to successfully detect falls and trigger immediate alarms. A cost-effective vision-based fall detection system that leverages advanced deep learning models and fusion methods to improve fall detection accuracy is proposed in [12]. The system includes object detection, pose estimation, action recognition, and result fusion, with probabilistic fusion demonstrating a significant performance improvement, achieving an average 0.84% increase in accuracy on the HAR-UP dataset.

Deep variational autoencoders (VAEs) are useful for time series anomaly detection because of their unsupervised training and uncertainty estimation abilities. A frequency-enhanced conditional variational autoencoder was introduced by [23] to capture complex temporal patterns by combining local and global frequency data to achieve enhanced anomaly detection. A novel generative framework that integrates VAEs with self-supervised learning was introduced in [25] to address data scarcity concerns and improve anomaly detection.

Human pose estimation techniques have also contributed significantly to fall detection. The work in [18] leverages the Mediapipe human pose estimation architecture, resulting in improved accuracy and real-time processing capabilities. Despite the efficiency of these systems, there are still limitations in dealing with dynamic environments and optimizing performance for real-time applications, which our work addresses.

3 Proposed Methodology

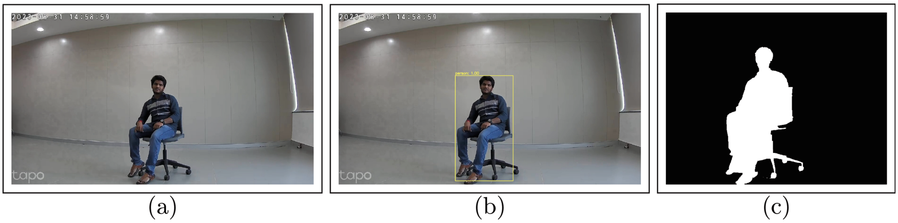
The proposed approach section will cover the major contributions of the paper. The dataset was collected to train and test the LSTM-VAE network. A customized optimization technique leverages the AMSGrad optimizer to overcome the convergence problem when stuck at local minima. The implementation of approaches such as gradient clipping and gradient checkpointing contributes to handling outliers.

3.1 Dataset Collection Protocol

The dataset was collected from 20 subjects who performed 80 activities of daily living (ADL) and 70 fall activities. To develop a framework robust in distinguishing falls from fall-like activities, the participants were made to perform five different activities: walking, falling from walking, running, sitting on a chair, and falling from a chair. There is minimal variance in the activities recorded. Each participant performed three trials for every activity. The system was trained on 70 ADL activities and tested on a combination of 10 ADL and 70 fall activities.

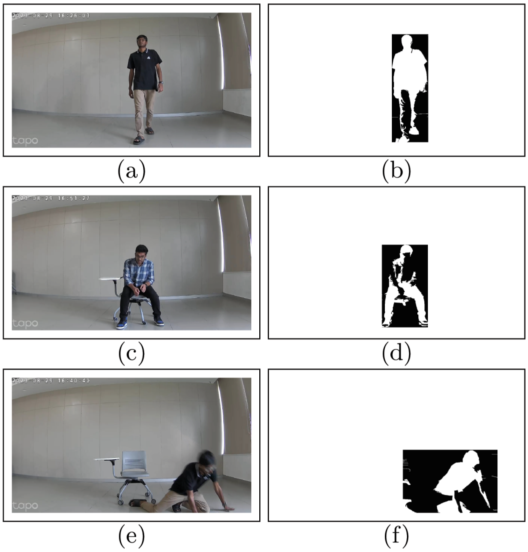
3.2 Data Pre-processing

The pre-processing section establishes a privacy-preserving mechanism for human activity recognition. The videos are divided into frames, and each frame undergoes a series of pre-processing operations. Each frame is normalized to a range between $[0, 1]$. To identify people in the video frames, sophisticated object detection algorithms like You Only Look Once (YOLO) [17] are first used. YOLO, a deep learning architecture pre-trained on the large ImageNet dataset [4], provides high accuracy in identifying a wide range of objects, including humans. After detecting humans, the human silhouettes are extracted from the detected regions using a contour-based approach. This method gathers the critical spatial information required for subsequent activity recognition challenges. The extracted human silhouettes undergo a series of morphological processes, such as erosion and dilation, to conceal identifiable features while retaining activity-related information. This method ensures that everyone recorded in the video feed-especially those involved in fall detection scenarios-remains anonymous, addressing privacy concerns related to ongoing monitoring and analysis. The procedure for obtaining a de-identified human image is given in Algorithm 1, and a pictorial representation of the process is depicted in Fig. 2. One potential failure case in the human de-identification process is the inaccurate masking of the human figure due to certain environmental conditions. Some of the failure cases are depicted in Fig. 3.



(a) (b) (c)

Fig. 2. Privacy-preserving for fall detection workflow : (a) represents the original image, (b) represents the person detection with YOLO, (c) represents the silhouette extraction and privacy protection



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Some cases where the de-identification process failed

Algorithm 1 The procedure for Privacy-Protection Mechanism

```
for each frame in the frames_of_a_video do
    Normalize each frame to obtain pixel values of the range [0, 1].
    $Normalized\_frame = frame/255$
    Apply pre-trained human detection algorithm
    Extract human regions
end for
for each frame that has extracted human regions do
    Identify the largest contour and extract the silhouette
    Apply erosion to obtain a fine contour
    Mask the original human with the obtained silhouette
end for
The fall detection process is carried out using the privacy-protected frames.
```

3.3 LSTM-Based Variational Autoencoder Network

An LSTM-VAE network is designed in the proposed fall detection system. The network is trained and reconstructed on normal activities. Hence, during the testing process, a high deviation in the reconstruction error is identified as a fall. An illustration of the LSTM-VAE network is shown in Fig. 4.

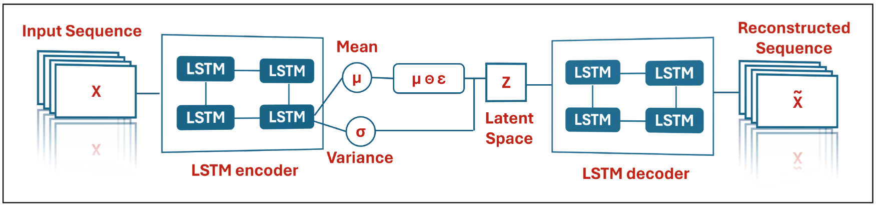


Fig. 4. Illustration of the LSTM-VAE Network for Fall Detection

Encoder. The encoder consists of a sequence of LSTM layers that process the input video frames sequentially. The LSTM layers learn significant representations of video sequences from the temporal dependencies captured in the data. These representations are then compressed into probabilistic interpretations, such as mean μ and standard deviation σ. A sample latent space Z is computed by adding noise ϵ to μ and σ using Eq. (5).

$$z = z_{\mu} + \epsilon \cdot e^{\frac{1}{2} z_{log_{\sigma}}} \tag{5}$$

where ϵ is drawn from a standard normal distribution $N(0, 1)$.

Decoder. The decoder reconstructs the sample latent space Z passed from the encoder network. The decoder, like the encoder, consists of LSTM layers but operates in reverse order. The reconstructed data is then compared with the original data using a binary cross-entropy (BCE) loss function, as given in Eq. (6). Additionally, Kullback-Leibler (KL) divergence loss is computed to measure the deviation between the prior and current distributions, as shown in Eq. (7).

$$\text{BCE Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right) \tag{6}$$

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2}\sum_{j=1}^{J}\left(1+\log(\sigma_j^2)-\mu_j^2-\sigma_j^2\right) \tag{7}$$

Optimization. Conventionally, the AMSGrad optimizer is used for non-stationary objectives. However, in the proposed framework, a smoothened AMSGrad optimizer is introduced to reduce noise. This is achieved by incorporating a smoothing parameter (ρ) that smooths the historical maximum of squared gradients. Additionally, a gradient clipping mechanism is imposed to prevent gradients from growing exponentially by applying a gradient threshold. The algorithm for the smoothened AMSGrad optimizer is provided in Algorithm 2. Initially, the hyperparameters β_1, β_2 that control the decay rate of the first and second moment estimates are initialized, along with ϵ to avoid division by zero, ρ as the smoothing parameter, and *clip_threshold* for the gradient clipping mechanism. The exponential moving average (exp_avg), exponential moving average of squared gradients (exp_avg_sq), and maximum exponential moving average of squared gradients (max_exp_avg_sq) are initialized to zero. In each iteration, these variables are updated, corrected for bias, and the smoothened maximum exponential moving average of squared gradients (smoothed_max_exp_avg_sq) is calculated by applying the ρ smoothing factor. Finally, every parameter and step size are updated with bias correction.

$$g = g * \frac{clip\_threshold}{norm(g)} \tag{8}$$

$$step\_size = \frac{lr}{\sqrt{(}smoothed\_max\_exp\_avg\_sq) + \epsilon} \tag{9}$$

$$p = \frac{p - step\_size * corrected\_exp\_avg}{\sqrt{(}smoothed\_max\_exp\_avg\_sq) + \epsilon} \tag{10}$$

The LSTM-VAE is trained on normal activities and tested on fall activities. If the reconstruction error during testing exceeds a predetermined threshold, the activity is flagged as an anomaly and marked as a fall. The threshold is dynamically computed using a moving average thresholding technique.

Initially, a moving average is computed within a sliding window of size 64. For each step, the average of the data points within the window is calculated, and then the window is advanced by one step. After the moving average is obtained,

Algorithm 2 Smoothened AMSGrad Optimizer Algorithm:

```
Initialize Hyperparameters : $\beta1, \beta2, \epsilon, \rho, clip\_threshold$
for each parameter $p$ do
    initialize $exp\_avg$, $exp\_avg\_sq$, $max\_exp\_avg\_sq$
    initialize $step = 0$
end for
for each iteration do
    compute gradients and L2 norm
    Apply gradient clipping function given in Equation (8) if $norm > clip\_threshold$
    Update variables:
    $exp\_avg = \beta1 * exp\_avg + (1 - \beta1) * gradient$
    $exp\_avg\_sq = \beta2 * exp\_avg\_sq + (1 - \beta2) * gradient^2$
    $max\_exp\_avg\_sq = max(max\_exp\_avg\_sq, exp\_avg\_sq)$
    for every bias correction do
        $corrected\_exp\_avg = exp\_avg/(1 - \beta1^{step})$
        $corrected\_exp\_avg\_sq = exp\_avg\_sq/(1 - \beta2^{step})$
        $smoothed\_max\_exp\_avg\_sq = \rho * smoothed\_max\_exp\_avg\_sq + (1 - \rho) * max\_exp\_avg\_sq$
        Update parameters: step_size and p using Equation (9) and (10) respectively.
    end for
end for
```

the deviation of each data point from the moving average is calculated. The threshold is determined by adding a margin, adjusted by the standard deviation, to the mean of the moving averages. An activity is identified as an anomaly if its deviation from the moving average exceeds the set threshold. This process is detailed in Algorithm 3.

4 Ablation Study

In this section, the ablation study conducted on the custom dataset for anomaly detection is discussed. Table 1 presents a comparative analysis of the current and smoothed AMSGrad optimizers. The performance of various fall detection techniques on our dataset is also covered in the table. Subsequently, Table 2 lists the performance of different object detection models, such as YOLO and MobileNet. It is observed that the YOLO model has achieved significant performance.

5 Experimental Results

In this section, the experimental results that were carried out on the custom dataset for anomaly detection are discussed. Our proposed approach uses an LSTM-VAE for detecting falls by analyzing latent feature distributions. By training the VAE on normal activities, the model learns a compact representation of typical human behavior. This involves capturing the regular patterns and variations in normal activities. Fall scenarios are different from normal activities and

Algorithm 3 Moving Average Thresholding

```
Require: Reconstruction errors, window size, threshold margin
Ensure: Anomalies identified based on reconstruction errors exceeding the threshold
Initialize empty list for moving averages: moving_averages
for $i$ in range(len(reconstruction_errors)) do
    $start\_idx \leftarrow \max(0, i - \text{window\_size} + 1)$
    $window \leftarrow$ reconstruction_errors[$start\_idx : i + 1$]
    $moving\_avg \leftarrow \frac{\sum_{j \in \text{window}} j}{\text{len}(window)}$
    Append $moving\_avg$ to moving_averages
end for
Calculate threshold: threshold $\leftarrow$ mean(moving_averages) + threshold_margin $\times$ std(moving_averages)
Initialize empty list for anomalies: anomalies
for each $error, moving\_avg$ in zip(reconstruction_errors, moving_averages) do
    if $error >$ threshold then
        Append $(error, moving\_avg)$ to anomalies
    end if
end for
return anomalies
```

Table 1. System performance employing different approaches using a smoothened AMSGrad optimizer.

Model	AMSGrad Optimizer		Smoothened AMSGrad Optimizer	
	Accuracy (%)	Train Time (h)	Accuracy (%)	Train Time (h)
CNN	87.25	8	90.62	6.9
LSTM	90.42	10	93.49	9.8
Autoencoder	90.98	7	92.52	5.2
LSTM-Autoencoder	95.57	10	99.23	7.2

Table 2. Comparison of the performance of one-stage object detection models using various approaches.

Model	MobileNet	YOLO
	Accuracy (%)	Accuracy (%)
CNN	90.89	90.62
LSTM	91.27	93.49
Autoencoder	94.98	92.52
LSTM-Autoencoder	98.57	99.23

often exhibit abrupt and significant changes in posture and motion. The VAE's latent space is designed to capture these subtle differences, making it effective for detecting falls as deviations from learned normal activity patterns. The latent space representation of normal activities and fall scenarios can be analyzed to illustrate why the approach is effective. Figures 5 and 6 shows a sample of latent space mean and latent space variance for normal and fall scenarios. The latent space visualizations shown in Fig. 7 reveal that the clusters for normal activities and fall activities are distinct, demonstrating that the model effectively separates these classes. The Gaussian-like distribution of the latent variables further supports that the model generalizes well to unseen data, thus preventing overfitting. These observations confirm that our approach is both effective and robust for fall detection.

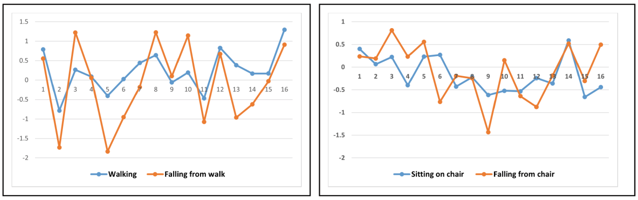


Fig. 5. Mean of the latent space for fall and normal scenarios

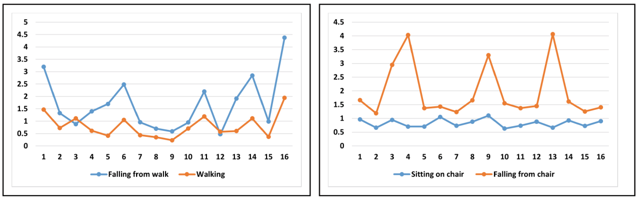


Fig. 6. Variance of the latent space for fall and normal scenarios

In the proposed framework, fall activities are considered as anomalies while the remaining activities are normal. The LSTM-VAE network is trained on the normal activities and tested on the fall activities. The reconstruction rate for

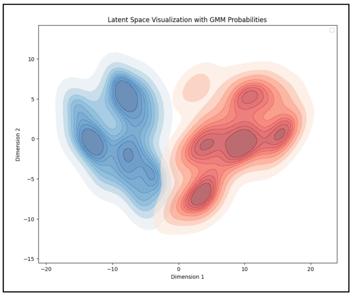


Fig. 7. Latent space visualization with GMM probabilities

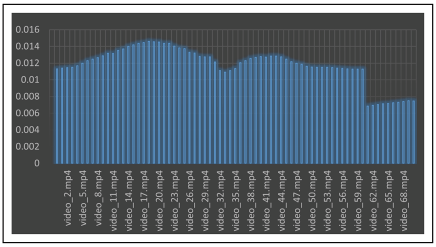


Fig. 8. The reconstruction error for video anomaly detection

normal activities is identified as the reconstruction threshold. Any activity that goes beyond the reconstruction threshold is categorized as a fall. The reconstruction error for all the videos in the test set is shown in Fig. 8.

The test dataset consists of 70 fall activities and 10 normal activities. It can be observed from the Fig. 8 that fall activities are clearly distinguished from normal activities based on the reconstruction threshold. The system achieved an accuracy of 99%, precision of 97%, and sensitivity of 99%.

To further evaluate the system, the UPfall dataset, and the URfall datasets are also tested. An accuracy of 100% was achieved on the URfall dataset. The results of the system are given in Table 3.

Table 3. Performance evaluation of the proposed system on benchmark fall detection datasets

	Accuracy (%)	Precision (%)	Recall (%)
Custom Dataset	99.23	97.10	99.34
UPfall dataset	99.42	97.27	97.27
URfall dataset	**99.98**	**99.21**	**99.35**

Some similar approaches in the literature are identified in the works discussed in [5,8,16,21,26]. The work discussed in [21] proposed a temporal convolutional hourglass autoencoder. A stream of images is fed into the temporal convolutional hourglass encoder to produce a compressed sample latent space. The temporal convolutional hourglass decoder reconstructs the sample, and any significant deviation from the input stream is considered an anomaly. The work proposed in [16] presents a DeepFall framework for fall detection using depth and thermal images. A OneFall generative adversarial network was proposed by [5]. This approach considers falls as anomalies and trains only on ADL, achieving an accuracy of 98.75% on the UPFall dataset. An activity recognition and fall detection network (ARFDNet) was proposed in [26]. They applied pose estimation to identify human keypoints, which were then fed into a sequence of convolutional neural networks and gated recurrent units for fall analysis. Their system achieved an accuracy of 96.7% on the UPFall dataset. Pose estimation and keypoint identification were also employed in the work proposed in [8]. The identified keypoints were fed into a CNN to identify spatial dependencies and then moved to an LSTM network to identify temporal dependencies. The system was evaluated on the UPFall dataset, achieving an accuracy of 98.59%. A comparison of these existing approaches with the proposed system is given in Table 4.

Table 4. Comparing fall detection methods against state-of-the-art approaches

	Accuracy (%)	Precision (%)	Sensitivity (%)
Scheme [21]	98.22	96.45	97.12
Scheme [16]	95.33	89.13	92.25
Scheme [5]	93.10	91.29	94.17
Scheme [26]	95.89	89.68	92.63
Scheme [8]	98.59	91.72	94.21
Our Proposed Work	**99.98**	**99.21**	**99.35**

6 Conclusion

In this work, we have addressed the challenge of limited video-based fall data by introducing a custom dataset comprising 80 activities of daily living (ADL) and 70 fall activities. To tackle privacy concerns, we have implemented a privacy-preserving mechanism that ensures anonymity in video footage. We designed a custom LSTM-VAE architecture to enhance fall detection accuracy, incorporating a gradient clipping mechanism and a smoothened variant of the AMSGrad optimizer to ensure smooth gradient convergence.

The system effectively distinguishes between normal and fall activities, achieving remarkable performance metrics: 100% accuracy, 99% precision, and 99% sensitivity on the URFall dataset. Future work could focus on further improving privacy-preserving methods by integrating radar-based detection systems with vision-based approaches, offering enhanced privacy and robustness in fall detection scenarios.

References

1. Adhikari, K., Bouchachia, H., Nait-Charif, H.: Activity recognition for indoor fall detection using convolutional neural network. In: 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pp. 81–84. IEEE (2017)
2. Alanazi, T., Babutain, K., Muhammad, G.: Mitigating human fall injuries: a novel system utilizing 3d 4-stream convolutional neural networks and image fusion. Image Vis. Comput., 105153 (2024)

3. Aslam, T., Harun, F.B., Ramli, A.F., Kadir, K.A., Nordin, M.N.: Deep learning based fall detection system. In: 2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), pp. 42–47. IEEE (2023)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
5. Galvão, Y.M., Portela, L., Barros, P., de Araújo Fagundes, R.A., Fernandes, B.J.: Onefall-gan: a one-class gan framework applied to fall detection. Int. J. Eng. Sci. Technol. **35**, 101227 (2022)
6. Gunale, K., Mukherji, P.: Indoor human fall detection system based on automatic vision using computer vision and machine learning algorithms. J. Eng. Sci. Technol. **13**(8), 2587–2605 (2018)
7. Hassan, E., Shams, M.Y., Hikal, N.A., Elmougy, S.: The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimed. Tools Appl. **82**(11), 16591–16633 (2023)
8. Inturi, A.R., Manikandan, V., Garrapally, V.: A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network. Arab. J. Sci. Eng. **48**(2), 1143–1155 (2023)
9. Kakara, R.: Nonfatal and fatal falls among adults aged 65 years-united states, 2020–2021. MMWR. Morbidity and Mortality Weekly Report **72** (2023)
10. Kepski, M., Kwolek, B.: Embedded system for fall detection using body-worn accelerometer and depth sensor. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), vol. 2, pp. 755–759. IEEE (2015)
11. Keskes, O., Noumeir, R.: Vision-based fall detection using st-gcn. IEEE Access **9**, 28224–28236 (2021)
12. Kim, J., Kim, B., Lee, H.: Fall recognition based on time-level decision fusion classification. Appl. Sci. **14**(2), 709 (2024)
13. Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y.: Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Health Inform. **18**(6), 1915–1922 (2014)
14. Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., Peñafort-Asturiano, C.: Up-fall detection dataset: a multimodal approach. Sensors **19**(9), 1988 (2019)
15. Mustapha, A., Mohamed, L., Ali, K.: Comparative study of optimization techniques in deep learning: application in the ophthalmology field. J. Phys. Conf. Ser. **1743**, 012002 (2021)
16. Nogas, J., Khan, S.S., Mihailidis, A.: Deepfall: non-invasive fall detection with deep spatio-temporal convolutional autoencoders. J. Healthc. Inform. Res. **4**(1), 50–70 (2020)
17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
18. Sengar, S.S., Kumar, A., Singh, O.: Efficient Human Pose Estimation: Leveraging Advanced Techniques with Mediapipe (2024). arXiv preprint arXiv:2406.15649
19. Tong, C., Tailor, S.A., Lane, N.D.: Are accelerometers for activity recognition a dead-end? (2020)
20. Tran, T.H., et al.: A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 1947–1952. IEEE (2018)

21. Wahla, S.Q., Ghani, M.U.: Visual fall detection from activities of daily living for assistive living. IEEE Access (2023)
22. Wang, Y., Lacotte, J., Pilanci, M.: The hidden convex optimization landscape of two-layer relu neural networks: an exact characterization of the optimal solutions (2020). arXiv preprint arXiv:2006.05900
23. Wang, Z., et al.: Revisiting VAE for unsupervised time series anomaly detection: a frequency perspective. In: Proceedings of the ACM on Web Conference 2024, pp. 3096–3105 (2024)
24. (WHO), W.H.O. https://www.who.int/news-room/fact-sheets/detail/ageing-and-health
25. Wu, Z., Cao, L., Zhang, Q., Zhou, J., Chen, H.: Weakly Augmented Variational Autoencoder in Time Series Anomaly Detection (2024). arXiv preprint: arXiv:2401.03341
26. 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. Knowl. Based Syst. **239**, 107948 (2022)

Pedestrian Trajectory Prediction Based on Improved Diffusion with Fourier Embeddings

Boyang Fu(✉), Jiashu Liao, Yixuan Yang, and Victor Sanchez

The University of Warwick, Coventry, UK
{boyang.fu,jiashu.liao,Yixuan.Yang.1,v.f.sanchez-silva}@warwick.ac.uk

Abstract. Predicting stochastic pedestrian trajectories is a complex task, requiring the integration of contextual information and the inherent uncertainty of human movement. Conventional generative models attempt to capture this uncertainty by mapping randomness to a latent space, producing a multimodal distribution of potential trajectories. These models, however, often fall short in complex scenarios with highly uncertain trajectories due to inadequate temporal dependency modeling. To address this shortcoming, we propose a framework that uses an improved conditional diffusion model that significantly enhances stochastic trajectory prediction. By conditioning on past trajectory data, the model iteratively adds Gaussian noise and employs a reverse generative process to output a diverse set of future trajectories. A novel denoising component merges noised predictions with historical data through a feature extractor, leveraging cross-attention mechanisms to intertwine past and future trajectories effectively. Furthermore, we enrich the framework's temporal analysis with Fourier embeddings, improving its time-series predictive power. Rigorous benchmarking on the ETH, UCY, and SDD datasets confirms that our framework outperforms several state-of-the-art methods in generating accurate future trajectories.

Keywords: Multimodal Trajectory Prediction · Diffusion Model

1 Introduction

Pedestrian trajectory prediction aims to forecast the future movement path of a pedestrian by analyzing the surrounding vehicles, other pedestrians, and the environment. Recently, through deep learning methods, pedestrian trajectory prediction has played a crucial role in tasks like autonomous driving [16], surveillance by drones [28], and robotics [34]. The main challenge of the pedestrian trajectory prediction task is the randomness and probabilistic nature of human movements. The majority of current solutions are based on deterministic predictions, where a single trajectory is predicted for each pedestrian by using a sequence-to-sequence structure, such as Recurrent Neural Network (RNN)-based [19,25] or Transformer-based [39] autoencoders. In this case, the

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 415–431, 2025.
https://doi.org/10.1007/978-3-031-78444-6_27

encoder extracts spatial and temporal information and the decoder predicts future trajectories. However, such methods are limited by social interactions and the uncertainty of human movement, hindering their performance. To accurately capture the uncertainty of future trajectories, we need a prediction system that generates an unbiased distribution of plausible future trajectories. Therefore, current research predominantly utilizes generative models [10,11,36] to learn the distribution of trajectories; i.e., to provide stochastic trajectory predictions.

In stochastic trajectory prediction, generative models conditioned on past trajectories are the most commonly used solutions. For example, methods employing explicit density functions, such as Conditional Variational Autoencoders (CVAEs), assume the outputs follow a Gaussian distribution to simulate future trajectories [17,22,32,36]. However, the expressiveness of their latent representations is limited, failing to capture all the subtle nuances of the data. As a result, the output often appears unnatural or overly simplified. Generative Adversarial Networks (GANs) are not confined by a fixed density form, thus offering greater flexibility and diversity in generating trajectories. However, GANs also face challenges during training, such as mode collapse and unstable gradients [2,7,21].

Recently, advancements in image generation [8,30], video synthesis [13,37], and audio synthesis [6,20] have led to the development of a new type of generative model: Denoising Diffusion Probabilistic Models (DDPMs) [14]. These models reduce prediction uncertainty effectively by injecting random noise into future trajectories to generate multiple plausible paths, learning the data generation process based on noise and the true data distribution. During inference, such models start with Gaussian noise and refine the noise samples through an iterative process. For example, MID [10] employs a Transformer-based vanilla diffusion model that has been shown to predict trajectories in a stochastic or deterministic manner. However, MID still faces challenges in integrating conditional information effectively, which limits its effectiveness in complex scenarios. Additionally, the inference process is time-consuming, rendering it unsuitable for applications that require immediate responses. To address these challenges, we propose a new predictive Denoiser capable of merging conditional information with noisy trajectories to effectively learn the distribution patterns of trajectories. Additionally, we leverage Denoising Diffusion Implicit Models (DDIM) to accelerate the sampling process, enabling faster generation of probabilistic trajectory distributions. The contributions of our work are as follows:

- We propose a novel Denoiser capable of effectively processing noise-injected trajectories and integrating past information through a cross-attention mechanism. This strategy allows the generated trajectories to be consistent with both historical behavior patterns and potential future changes.
- We incorporate the Fourier transform to enhance our framework's analytical capabilities for time-series data. This allows capturing and predicting pedestrian movements across different time scales, further improving performance.
- Compared with the recent baselines, our framework showcases outstanding performance in evaluations on the pedestrian trajectory prediction benchmarks, ETH, UCY and SDD.

2 Related Work

Early research in trajectory prediction primarily focuses on deterministic prediction, which involves using historical trajectory data to predict a set of positions for each of several pedestrians at future time steps. The time-series nature of this task makes the use of RNNs and autoencoders in a sequence-to-sequence structure particularly suitable. For example, Social-LSTM [1] introduces a social pooling structure that integrates the hidden states of each pedestrian to capture and understand the interactions and relationships between moving objects. Sophie [31] employs the VGG-19 network to extract features from aerial views and introduces physical and social attention mechanisms. With the advent of the Transformer, multi-head attention is more suited for processing sequential data. For instance, STAR [38] employs a Transformer to capture spatial and temporal information, which is subsequently combined with a graph structure to achieve complex spatio-temporal interactions. This Transformer-based graph convolution mechanism is used to dynamically learn and adjust to the complex spatial relationships between nodes in the graph, while separate temporal transformers are used to capture the temporal dependencies between graphs. The recently proposed EqMotion [35], which uses interaction graphs, ensures motion equivariance under Euclidean geometric transformations and invariance of agent interactions, providing robust and accurate multi-agent motion predictions.

Recently, there has been a significant focus on stochastic trajectory prediction to predict a set of plausible trajectories given past trajectories, also known as multimodal trajectory prediction. Solutions based on Distributed Generative Models (DGMs) are effective in generating diverse trajectories that accurately reflect real-world complexity by leveraging probabilistic frameworks. GANs and CVAEs are particularly suited in this case, with GANs emphasizing diversity. For example, the combination of Bicycle-GAN and Graph Attention Networks [21] explore multimodal trajectory prediction by accounting for social interactions. MG-GAN [7] enhances the diversity and realism of trajectory prediction through a multi-generator strategy, effectively avoiding the generation of outlier samples. DESIRE [22] employs a CVAE for multi-trajectory prediction, leveraging prior and posterior distribution constraints to learn trajectory distributions. More recent research integrates contrastive learning into CVAEs [5,12], enhancing feature and pattern discrimination. Social-VAE [36] incorporates attention mechanisms for processing time-series data. It is important to note that CVAEs may fail at capturing complex data distributions due to assumptions made about the latent space structure, impacting trajectory generation quality.

Lately, the work in [10] introduces MID, an innovative diffusion model-based strategy that finely simulates the uncertainty of movement through a progressive noise addition process, achieving a transition from vague to precise trajectory prediction. BOsampler [3] introduces unsupervised sampling, which enhances the effectiveness of stochastic human trajectory generation. LED [27], on the other hand, uses a two-stage training process that first trains a complete denoiser followed by the fast sampling component. The unsupervised sampling technique, however, may encounter challenges with noise and variability in the trajectories without labeled data. Moreover, a two-state training process may be computationally complex and not fully leverage the denoiser's potential understanding of trajectory patterns. Our framework, which is based on a diffusion model and a single training stage, improves stochastic pedestrian trajectory prediction by introducing 1) a novel Denoiser that enhances past information integration, 2) Fourier embeddings that enhance spatio-temporal analysis capabilities, and 3) DDIM to accelerate inference when sampling the plausible future trajectories. Our framework hence enhances adaptability to real-time scenarios.

3 Proposed Framework

The goal of stochastic pedestrian trajectory prediction is to forecast future trajectories based on given historical information and the surrounding environment. The historical trajectory of pedestrian i is represented by $P_i = [p_1^i, p_2^i, \ldots, p_t^i] \in \mathbb{R}^{t\times 2}$, where t denotes the length of the observation time steps, and $p_t^i \in \mathbb{R}^2$ indicates the pedestrian's $\{x, y\}$ position at time t. The historical trajectory information of the j^{th} neighbor is denoted as $N_j^i = [N_{j,1}^i, N_{j,2}^i, \ldots, N_{j,t}^i] \in \mathbb{R}^{t\times 2}$, where $N_{j,t}^i \in \mathbb{R}^2$ describes the $\{x, y\}$ position of the neighbor at time t. We define the future trajectory of pedestrian i as $F_i = [f_{t+1}^i, f_{t+2}^i, \ldots, f_{t+T}^i] \in \mathbb{R}^{T\times 2}$, where T is the time step length of the future trajectory. The task's goal is to learn the conditional probability $p_\theta(F_i \mid P_i, \{N_j\})$, which represents the probability distribution of a pedestrian's future trajectory with given historical trajectories.

3.1 Framework Overview

Figure 1 illustrates our framework, which predicts future trajectories through a diffusion process, using past trajectory information as a conditional input. During the training phase, noise is added to ground truth future trajectories over $k \in [1, K]$ time steps. The denoising network is trained to predict the noise and distill temporal features from the trajectory data, enabling it to reconstruct future paths more accurately. During inference, our framework generates initial trajectory estimates from a Gaussian distribution, incorporates past trajectories as context, and progressively refines the predicted trajectory using the Denoiser. This iterative refinement process allows the framework to effectively navigate the inherent uncertainties within trajectory forecasting, yielding more precise predictions.

3.2 Extracting Conditional Information

Figure 2 shows the architecture of our Denoiser. Inspired by [32], the conditional information is extracted by a Social-Temporal Encoder, which captures and encodes the social interactions among pedestrians; i.e., the historical trajectories of the target and its neighbors. Our Denoiser comprises two components: the historical information temporal encoder, $\phi(\cdot)$, and the neighboring information social encoder, $\psi(\cdot)$. The temporal encoder, $\phi(\cdot)$, uses a Long Short-Term

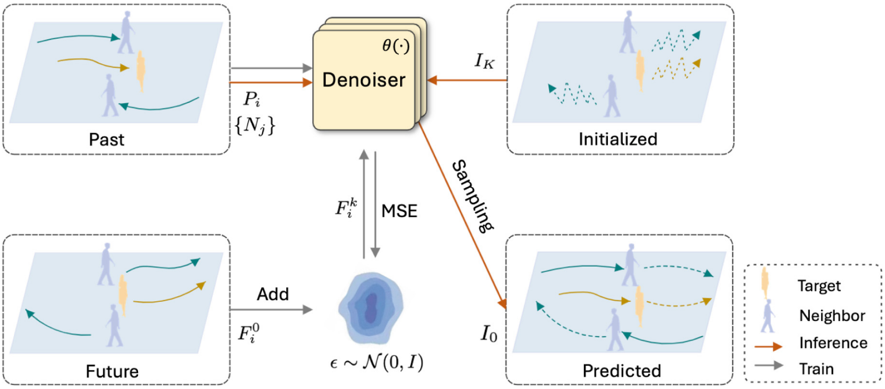


Fig. 1. Overview of the proposed framework. During the training phase, the past trajectories of the target pedestrian P_i and set of neighbors $\{N_j\}$ serve as conditional information guiding the diffusion learning process. The model trains the Denoiser $\theta(\cdot)$ to predict noise and restore future trajectories by iteratively adding noise $\epsilon \sim \mathcal{N}(0, I)$ to the target's future trajectory, i.e., from F_i^0 to F_i^k, over $k \in [1, K]$ time steps. In the inference phase, the trajectory I_K is initialized from Gaussian distribution combined with conditional information and refined through K iterations of the Denoiser, by sampling and predicting noise to produce the predicted trajectory I_0.

Memory (LSTM) network [15] with the $\{x, y\}$ position and speed of pedestrian i to produce the feature p_i^ϕ of dimension h. The social encoder, $\psi(\cdot)$, also employs an LSTM to encode the historical information of the neighbors associated with pedestrian i. Since the number of neighbors and their interactions can fluctuate over time, the social encoder $\psi(\cdot)$ encodes the past trajectories of all neighbors within a radius r by using a single-layer LSTM. Subsequently, an attention mechanism is applied to calculate the influence of each neighbor on the target pedestrian. This results in an aggregated feature set, denoted by $p_t^\psi \in \mathbb{R}^h$. These aggregated features encapsulate both the positional and velocity data of all neighbors within the pedestrian's vicinity, providing a comprehensive social context for trajectory prediction. After the encoding step, the historical features p_i^ϕ and the neighboring features p_t^ψ are concatenated to create the conditional pedestrian features $\mathbf{P}_i \in \mathbb{R}^{2h}$.

3.3 Forward Process

During training, our framework uses a diffusion model in which forward diffusion obfuscates the ground truth future trajectories, F_i, by incrementally adding noise over $k \in [1, K]$ time steps to produce the noisy trajectory $F_i^k \in \mathbb{R}^{T \times 2}$:

$$F_i^k = \sqrt{\alpha_k} F_i^0 + \sqrt{1 - \alpha_k}\epsilon, \quad k = 1, 2, \ldots, K \tag{1}$$

where F_i^0 represents the initial future trajectory at the starting point, the value of $k \in [1, K]$ is randomly selected, and $\epsilon \sim \mathcal{N}(0, I)$ is a noise matrix matching F_i^0's

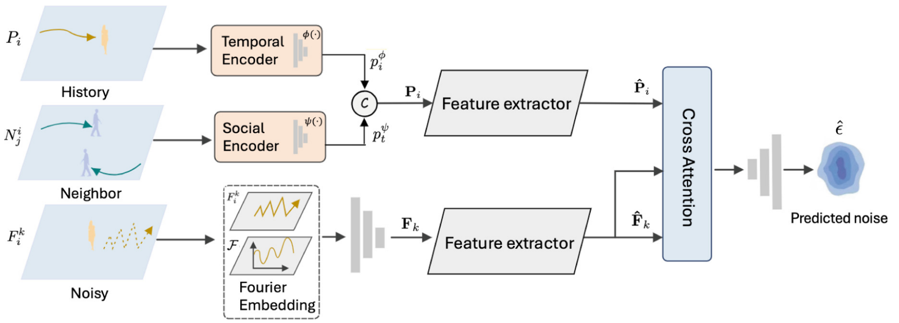


Fig. 2. The Denoiser. The historical trajectory features of the target p_i^{ϕ} and neighbors p_t^{ψ} are extracted by the temporal encoder $\phi(\cdot)$ and social encoder $\psi(\cdot)$, respectively. The noisy trajectory F_i^k is concatenated with the Fourier embeddings $\mathcal{F}$ and fed into a fully connected layer to produce $\mathbf{F}_k$. Subsequently, after the feature extractor, the temporal features $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{F}}_k$ are merged through cross attention and ultimately decoded into the predicted noise $\hat{\epsilon}$.

dimensions. Note that the variable α_k is a pre-determined coefficient controlling the amount of noise added for the selected value of k. The value of α_k decreases as k increases, i.e., as more steps are used, more noise is added to the trajectory. This method allows our framework to learn the features of trajectories under different noise levels, which in turn facilitates the reconstruction of the original trajectory during the reverse diffusion process. Such a forward diffusion process provides training data for the reverse process, enabling it to learn how to recover the original trajectory from noisy data.

3.4 Fourier Embedding

We introduce Fourier encoding to capture temporal features in the noisy trajectories F_i^k. This involves applying *sine* and *cosine* transformations, distributed logarithmically to achieve precise encoding for each dimension, thus effectively capturing periodic features across different time scales:

$$\mathcal{F}(F_i^k, \delta) = \begin{cases} \sin(2\pi \cdot 2^{m \cdot \delta} \cdot F_i^k) \\ \cos(2\pi \cdot 2^{m \cdot \delta} \cdot F_i^k), \end{cases} \tag{2}$$

where δ is a scaling factor that determines the granularity of the frequency distribution. In our model, the index m starts at 1 and increases to M, indexing different frequency bands used in the encoding process. Frequencies for these bands are logarithmically spaced, beginning at the lowest and increasing to a maximum determined by the power of two, which is scaled by δ and the band index m. The noisy trajectory data F_i^k and its corresponding Fourier-encoded features $\mathcal{F}(F_i^k, \delta)$, are fused along the feature dimension, creating an enhanced feature space $\hat{F}_k$:

$$\hat{F}_k = F_i^k \oplus \mathcal{F}(F_i^k, \delta). \tag{3}$$

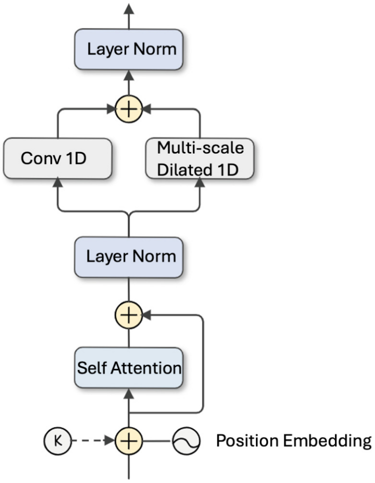


Fig. 3. Feature extractor architecture. The noisy trajectory $\mathbf{F}_k$ and condition information $\mathbf{P}_i$ are fed, separately, to the feature extractor for further feature extraction. K is a time encoding, which is only used on $\mathbf{P}_i$.

Such feature fusion further expands the framework's ability to exploit temporal features, allowing it to gain a more comprehensive understanding of temporal dynamics, enhancing the accuracy of complex time series behavior prediction. Following this, $\hat{F}_k$ is fed into a fully connected layer to produce $\mathbf{F}_k$ with dimensions $\mathbb{R}^{T\times 2h}$, preparing the data for subsequent feature extraction.

3.5 Feature Extractor

The *feature extractor* used by the Denoiser, as shown in Fig. 2, is detailed in Fig. 3. It extracts finer-grained features from both $\mathbf{F}_k$ and $\mathbf{P}_i$. A self-attention mechanism is first employed to process the input trajectory data, establishing direct dependencies between different positions within the sequence. Subsequently, both Conv1D convolutions and multi-scale dilated 1D convolutions are utilized for extracting local features and trend features, respectively. Finally, layer normalization is applied to unify the data distribution and enhance the framework's stability. The final output of the *feature extractor* is utilized in the subsequent denoising step. Each component of the *feature extractor* is detailed next.

Time Encoding: The *feature extractor* uses time encoding on $\mathbf{P}_i$ to amplify its capability in capturing time-steps of noise. Specifically, time encoding for k time steps is achieved as follows:

$$Emb(k) = concat(\beta_k, \sin(\beta_k), \cos(\beta_k)), \tag{4}$$

where k is the same value used to add noise to F_i, β_k is the scalar representation of the value of k, and the *concat* operation produces a unified time embedding vector.

Positional Encoding: A $2h$-dimensional position embedding p is also concatenated with $\mathbf{F}_k$ and the time-feature-injected trajectory $\mathbf{P}_i$. This embedding is computed as follows:

$$\begin{cases} p(i,2j) & = \sin\left(\frac{i}{10000^{2j/d}}\right) \\ p(i,2j+1) & = \cos\left(\frac{i}{10000^{2j/d}}\right), \end{cases} \tag{5}$$

where d denotes the dimension of the position embedding, which equals the dimension of F_k and that of P_i; i represents the time step; and $2j$ and $2j+1$ refer to the even and odd dimension indices within d. This guarantees a distinctive encoding for every position.

Self Attention: After concatenation with the time and position embeddings, $\mathbf{F}_k$ and $\mathbf{P}_i$ are fed into the self-attention module to compute $Q = XW^Q$, $K = XW^K$, $V = XW^V$, where X represents the input, $\{W^Q, W^K, W^V\}$ are the learnable weight matrices, and d_k is the dimensionality of the keys:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{6}$$

Note that Eq. 6 normalizes the dot product of Q and K to obtain attention weights, which are used to perform a weighted sum of V, yielding a weighted feature representation. Also, note that the output of the self-attention module is combined with its corresponding input to form a residual connection to mitigate gradient vanishing. The result is then standardized using *LayerNorm*, which enhances stability and convergence speed by standardizing features.

Local Feature Extraction: We employ a standard 1D convolutional layer to extract local features. This approach aims to improve the model's capacity to capture local temporal patterns within the sequence data.

Trend Feature Extraction: To capture intrinsic trend features within the trajectory data more comprehensively, we incorporate a multi-scale dilated 1D convolutional architecture. This architecture is adept at detecting varying trends and patterns across multiple time scales, thereby enriching the framework's interpretation of trajectories. Specifically, we employ 1D convolutions with dilation rates $\{1, 2, 4\}$, each with a kernel size set to 3, to capture and integrate multi-scale information. The outputs from the multi-scale dilated 1D convolutional architecture are added to those of the 1D convolution. This comprehensive feature representation allows the model to effectively capture and articulate both local and trend features at each time step in the trajectory data. The output of the *feature extractor block* is then $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{P}}_i$ (see Fig. 2).

3.6 Conditional Fusion

We use a cross-attention mechanism as a means of fusion. This approach enables the effective blending of information from future noisy trajectories with conditional past trajectories to guide the denoising process:

$$Cross\_Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{7}$$

where Q and K are derived from $\hat{\mathbf{F}}_k$, capturing querying and matching aspects of the future trajectory, respectively, while V encapsulates the conditional past trajectories $\hat{\mathbf{P}}_i$, providing the necessary context for denoising reconstruction. By employing this cross-attention mechanism, the framework selectively concentrates on relevant segments of the noisy future trajectories that hold the most promise for reconstructing the future trajectories under the guidance of the past trajectories. Finally, we feed the output of this cross-attention mechanism to a fully connected layer to compute the predicted noise $\hat{\epsilon} \in \mathbb{R}^{T\times 2}$.

3.7 Training

During training, given the condition and the noisy future trajectory, the Denoiser aims to estimate the noise ϵ that was added to the ground truth future trajectories. The loss function for the denoising model is defined as:

$$L(\theta) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I),F_i^0,k}\left\|\epsilon - \hat{\epsilon}_\theta(F_i^k, P_i, \{N_i^j\})\right\|, \tag{8}$$

where $\hat{\epsilon}_\theta(F_i^k, P_i, \{N_i^j\})$ is the noise estimated by the model with parameters θ, and F_i^k, P_i, and $\{N_i^j\}$ represent the input noisy trajectory, the target's past trajectory, and the neighboring trajectories, respectively.

Throughout the training process, the estimated noise $\hat{\epsilon}_\theta$ is continuously refined to enhance the framework's ability to reverse the diffusion process. The final model is expected to accurately predict the noise for any given noisy future trajectory through the condition P_i and $\{N_i^j\}$.

3.8 Inference

During the inference stage, the denoising process learned in the training phase is applied to generate clean trajectories from noisy data. We commence the inference with an initial noisy trajectory assumed to be the result of adding noise to a clean trajectory over K steps i.e., $I_i^K \sim \mathcal{N}(0, I)$. We then apply reverse diffusion over K steps to reconstruct the original trajectory. The trajectory at step $k-1$ in this reverse process can be expressed as:

$$I_i^{k-1} = \frac{1}{\sqrt{\alpha_k}}\left(I_i^k - \frac{\beta_k}{\sqrt{1-\alpha_k}}\hat{\epsilon}_\theta(I_i^k, P_i, \{N_i^j\})\right) + \sqrt{\beta_k}z, \tag{9}$$

where $z \sim \mathcal{N}(0, I)$ is a random variable sampled from a standard Gaussian distribution, significant for steps $k > 1$. Here, α_k and β_k are coefficients that modulate the diffusion process, $\hat{\epsilon}_\theta(I_i^k, P_i, \{N_i^j\})$ represents the noise estimated by the model with parameters θ, and $\sqrt{\beta_k}z$ accounts for the added noise at step k. The denoising network's input includes the current noisy trajectory I_i^k, the target's past trajectory P_i, and the neighboring trajectories $\{N_i^j\}$.

Note that the standard reverse DDPM sampling approach requires K denoising steps to generate a sample, which can be time-consuming and computationally expensive. To address this, we incorporate the DDIM [33] sampling technique, which skips every γ steps in the reverse process. This modification reduces the number of iterations to K/γ, effectively accelerating the sampling process by a factor of γ, thus leading to more efficient and expedient trajectory generation.

By incorporating the DDIM technique, we enhance efficiency and operational speed while maintaining the quality of the generated samples.

4 Experiments and Analysis

Datasets: We evaluate our framework on three benchmark datasets: ETH [4], UCY [23], and Stanford Drone Dataset (SDD) [29]. These datasets reflect real-world scenarios and record pedestrian movements and interactions from a bird's-eye view. The SDD dataset comprises 20 different scenes, the ETH dataset includes two scenarios (ETH and HOTEL), and the UCY dataset includes three scenarios (UNIV, ZARA1, and ZARA2). These scenarios were collected at a sampling rate of 2.5 Hz, showing rich multi-person interactions in unrestricted settings, allowing for various pedestrian paths and interaction patterns. For ETH and UCY, we employ a leave-one-out approach similar to other studies, training on four scenarios and testing on the remaining one. After observing the initial 8 frames (3.2 s), the task is to estimate the pedestrian's coordinates for the next 12 frames, i.e., over a window of 4.8 s.

Metrics: We use two primary metrics: Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE measures the average Euclidean distance between each point of the predicted trajectory and the corresponding point of the actual trajectory over the entire forecast period. FDE measures the Euclidean distance between the actual and the predicted positions at the final time step. These are mathematically expressed as:

$$ADE = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2}, \tag{10}$$

$$FDE = \sqrt{(x_T - \hat{x}_T)^2 + (y_T - \hat{y}_T)^2}, \tag{11}$$

where T denotes the total number of time steps and (x_t, y_t) and $(\hat{x}_t, \hat{y}_t)$ are the coordinates of the actual trajectory and the predicted trajectory, respectively, at time t. In light of the stochastic property of our framework, we follow other works [10] by adopting a Best-of-N strategy, selecting the best result out of $N = 20$ trials to calculate the final ADE and FDE.

Implement Details: Our framework is built on PyTorch 1.13.1 and trained end-to-end with no pre-training. The optimizer chosen is AdamW for the first 80% of the training and Adamax for the remaining 20%. The initial learning rate is set to 0.001, which is set to decay exponentially at rate of 0.98. Each batch size is set to 256. The LSTM hidden dimension for past trajectories is 128, and the number of channels for the dilated 1D convolution is 256 with a kernel size of 3. The embedding dimension for self-attention is 256 with 4 heads, while the dimension for cross-attention is 768. We use a value of $\delta = 0.5$ and $M = 4$ for the Fourier embeddings. Our experiments are conducted on an NVIDIA RTX-3060.

Table 1. Performance of several models on the ETH and UCY datasets in terms of ADE/FDE metrics with a Best-of-20 strategy. The best and second-best results are highlighted in **bold** font and underlined, respectively.

Model	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social-GAN [11]	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
SoPhie [31]	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.54/1.15
CGNS [24]	0.62/1.40	0.70/0.93	0.48/1.22	0.32/0.59	0.35/0.71	0.49/0.97
STGCNN-C [4]	0.64/1.00	0.38/0.45	0.49/0.81	0.34/0.53	0.32/0.49	0.43/0.66
MG-GAN [7]	0.47/0.91	0.14/0.24	0.54/1.07	0.36/0.73	0.29/0.60	0.36/0.71
PECNet [26]	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
CAGN [9]	0.41/0.65	0.13/0.23	0.32/0.54	0.21/0.38	0.16/0.33	0.25/0.43
MID-DDIM	0.41/0.70	0.19/0.32	**0.22/0.43**	0.26/0.51	0.17/0.34	0.25/0.46
MID* [10]	0.40/0.73	0.17/0.29	**0.22**/0.47	0.20/0.39	0.15/0.30	0.22/0.43
BOsampler [3]	0.52/0.95	0.19/0.39	0.30/0.67	**0.14**/0.33	0.20/0.45	0.27/0.56
LED [27]	0.39/**0.58**	**0.11/0.17**	0.26/**0.43**	0.18/**0.26**	**0.13/0.22**	**0.21/0.33**
Ours-DDIM	0.41/0.70	0.16/0.28	0.26/0.46	0.23/0.41	0.18/0.36	0.24/0.44
Ours (w/o Fourier)	0.39/0.70	0.15/0.24	0.24/0.45	0.21/0.41	0.17/0.30	0.23/0.41
Ours	**0.37**/0.61	0.12/0.20	0.23/0.44	0.20/0.36	0.15/0.29	**0.21**/0.38

* reproduced results from the open source implementation.

4.1 Comparison with the SOTA

In Table 1, we compare our framework with the state-of-the-art methods in [3,4,7,9–11,24,26,27,31] on the ETH & UCY datasets. Among them, Social-GAN [11], SoPhie [31], and MG-GAN [7] implement GAN-based generative methods, PECNet [26] utilizes a conditional VAE, PECNet and CAGN [9] are goal-conditioned methods, and MID [10], LED [27] employ diffusion models. The results reported for MID-DDIM, which uses DDIM sampling, are those obtained after reproducing this method. 'Ours-DDIM' refers to the case of using DDIM sampling with $K = 2$ steps, 'Ours' refers to using the standard sampling with $K = 80$ steps, and 'Ours w/o Fourier' refers to not using Fourier embeddings with standard sampling with $K = 80$ steps.

Compared to other models, our framework achieves strong performance on the majority of scenarios of the benchmarks and attains the best average ADE

Table 2. Performance of several models in terms of ADE/FDE metrics on the SDD dataset with a Best-of-20 strategy. The best and second-best results are highlighted in **bold** font and underlined, respectively.

Model	SDD
Social-GAN [11]	27.23/41.44
SoPhie [31]	16.27/29.38
CGNS [24]	15.60/28.20
MG-GAN [7]	13.60/25.80
PECNet [26]	9.96/15.88
GroupNet [9]	9.31/16.11
MID [10]	9.73/15.32
LED [27]	8.48/**11.66**
Ours	**8.21**/14.62

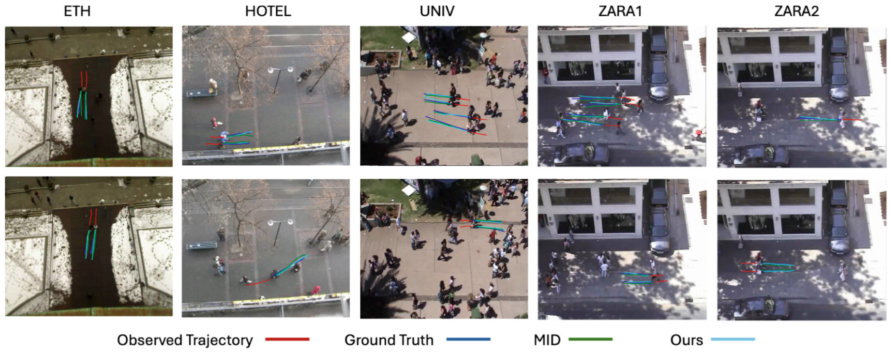


Fig. 4. Qualitative comparison of our framework against MID (baseline) on the ETH, HOTEL, UNIV, ZARA1, and ZARA2 scenarios.

and second-best average FDE. Note that although LED achieves better results in several of the scenarios, it is a two-stage training model: training the denoiser first, followed by training fast sampling process, whereas our model requires a single training stage. When we use the fast DDIM sampling, our framework also attains better results than the majority of evaluated methods; however, the best results are attained when standard sampling is used along with the Fourier embeddings. The results for the SDD dataset in Table 2 further confirm our framework's enhanced ability to capture the relationship between past and future trajectories, thereby better grasping the association of stochastic trajectories over time.

Figure 4 shows visual results for four scenes of the ETH, HOTEL, UNIV, ZARA1, and ZARA2 scenarios. The trajectories predicted by our framework (in light blue) are compared with the ground truth trajectories (in blue). Compared to the predictions made by MID (in green), which can be considered as a baseline, our framework can predict trajectories that are closer to the ground truth.

4.2 Ablation Studies

Fusion Strategies: As tabulated in Table 3, we investigate several fusion strategies for integrating conditional information from past trajectories with noisy trajectories. We use the five scenarios of the ETH &UCY datasets. Specifically, we replace the cross-attention mechanism in the Denoiser (see Fig. 2) with three different fusion approaches: Concatenation, Addition, and Gate Fusion. In the Gate Fusion strategy, a gating signal modulates the significance of features, allowing the model to dynamically adjust the degree of feature fusion based on the current context. The results indicate that cross-attention effectively retains essential trajectory features while filtering out noise across consecutive time steps. In contrast, the other strategies may lose information or introduce unnecessary noise when capturing the complex relationships of the trajectories.

Table 3. Average ADE & FDE values when using different fusion strategies on the five scenarios of the ETH & UCY datasets.

Fusion Module	ADE	FDE
Concatenation	0.22	0.40
Addition	0.22	0.41
Gate Fusion	0.23	0.43
Cross-attention	**0.21**	**0.37**

Table 4. Average ADE & FDE values when using different scaling factors (δ) and maximum frequencies (M) on SDD.

Scaling Factor	$\delta = 2$			$\delta = 1$			$\delta = 0.5$		
Maximum frequency	M = 4	M = 6	M = 8	M = 4	M = 6	M = 8	M = 4	M = 6	M = 8
ADE	8.29	8.37	8.43	8.31	8.59	8.36	**8.21**	8.47	8.38
FDE	14.79	14.84	15.62	14.63	15.30	14.78	**14.62**	14.84	14.60

Fourier Embedding: As shown in Table 4, we investigate different scaling factors δ and maximum frequencies M to evaluate the impact of the Fourier embeddings on performance on SDD. For $\delta = 2$, ADE and FDE values increase as M increases, suggesting that excessively high frequencies may introduce more noise, leading to performance degradation. When $\delta = 1$, the variations in ADE and FDE values are relatively small across different frequencies, but the performance is poor at $M = 6$, indicating that moderate frequencies may be less effective than higher or lower frequencies. For $\delta = 0.5$, the best results are achieved at $M = 4$, demonstrating that lower frequencies can better capture the dynamic characteristics of the trajectories at a fine granularity.

Sampling Methods: We investigate the performance of standard sampling, DDIM, UniPC [40], and K-diffusion [18] sampling across various time step settings and for several prediction windows. We use the five scenarios of the ETH &

Table 5. Average ADE/FDE values for several prediction windows when using different sampling methods on the ETH & UCY datasets.

Sampling method	Prediction window				Inference (ms)
	1.2 s	2.4 s	3.6 s	4.8 s	
Standard ($K = 100$)	0.04/0.04	0.09/0.14	0.15/0.25	0.21/0.39	$\sim$996
Standard ($K = 80$)	0.03/0.04	0.09/0.13	0.14/0.25	**0.21/0.38**	$\sim$974
Standard ($K = 50$)	0.07/0.08	0.15/0.21	0.24/0.38	0.34/0.55	$\sim$**495**
K-diffusion($K = 20$)	0.07/0.07	0.14/0.18	0.22/0.31	0.30/0.48	$\sim$213
K-diffusion($K = 5$)	0.06/0.06	0.12/0.16	0.19/0.29	**0.27/0.43**	$\sim$61
K-diffusion($K = 2$)	0.08/0.07	0.13/0.17	0.21/0.31	0.29/0.46	$\sim$**30**
Unipc ($K = 20$)	0.04/0.05	0.10/0.15	0.16/0.28	**0.24/0.44**	$\sim$117
Unipc ($K = 5$)	0.05/0.06	0.11/0.16	0.17/0.29	0.25/0.45	$\sim$48
Unipc ($K = 2$)	0.07/0.06	0.13/0.17	0.20/0.31	0.28/0.47	$\sim$**28**
DDIM ($K = 20$)	0.04/0.05	0.10/0.15	0.17/0.28	0.27/0.47	$\sim$207
DDIM ($K = 5$)	0.05/0.06	0.11/0.16	0.17/0.29	0.25/0.46	$\sim$59
DDIM ($K = 2$)	0.06/0.07	0.12/0.195	0.18/0.30	**0.24/0.44**	$\sim$**29**

UCY datasets. The average results are tabulated in Table 5, where the inference time is based on the analysis of a single trajectory on the ETH dataset. UniPC sampling introduces adjusted noise prediction and sample update mechanisms to smooth transitions between time steps, thereby enhancing the stability and quality of the generated samples. The K-diffusion sampling employs pseudo-Langevin Markov Sampling (PLMS) to increase efficiency. We observe that as the number of steps for standard sampling decreases from 100 to 50, the inference time significantly drops from $\sim$996 ms to $\sim$495 ms. However, there is a notable performance degradation at $K = 50$, with the optimal results achieved at $K = 80$. DDIM sampling with $K = 2$ markedly reduces the inference time to $\sim$29 ms, with relatively stable results. UniPC sampling also shows promising results, reducing the inference time to $\sim$28 ms at $K = 2$ while maintaining competitive performance. Similarly, K-diffusion sampling reduces the inference time to $\sim$30 ms at $K = 2$, with performance comparable to UniPC and DDIM. These findings suggest a trade-off between inference speed and prediction accuracy, which may be related to the degree of information loss during the sampling process. In standard sampling, more steps might help to maintain the continuity of predictions, while in DDIM, UniPC, and K-diffusion sampling, the rapid inference process does not significantly compromise the quality of the predictions.

5 Conclusion

In this paper, we proposed a framework for the stochastic prediction of pedestrian trajectories by using an improved diffusion model. By introducing a novel denoising component, our diffusion model effectively combines noise-injected trajectories with historical data, utilizing feature fusion to strengthen the linkage between past and future trajectories. Rigorous testing on the ETH, UCY, and SDD datasets showed that our framework surpasses several of the state-of-the-art methods in generating accurate pedestrian trajectory predictions, showcasing its potential and practical value in real scenarios.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese: social LSTM: human trajectory prediction in crowded spaces. In: CVPR, pp. 961–971 (2016)
2. Amirian, J., Hayet, J.B., Pettr, J.: Social ways: learning multi-modal distributions of pedestrian trajectories with gans. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition Workshops (2019)
3. Chen, G., Chen, Z., Fan, S., Zhang, K.: Unsupervised sampling promoting for stochastic human trajectory prediction. In: CVPR, pp. 17874–17884 (2023)
4. Chen, G., Li, J., Lu, J., Zhou, J.: Human trajectory prediction via counterfactual analysis. In: IEEE ICCV, pp. 9824–9833 (2021)
5. Chen, G., Li, J., Zhou, N., Ren, L., Lu, J.: Personalized trajectory prediction via distribution discrimination. In: IEEE International Conference on Computer Vision, pp. 15580–15589 (2021)
6. Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M.: Wavegrad: Estimating Gradients for Waveform Generation (2020). arXiv preprint arXiv:2009.00713
7. Dendorfer, P., Elflein, S., Leal-Taixé, L.: Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In: 2021 IEEE International Conference on Computer Vision, pp. 13158–13167 (2021)
8. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 8780–8794 (2021)
9. Duan, J., Wang, L., Long, C., Zhou, S., Zheng, F., Shi, L., Hua, G.: Complementary attention gated network for pedestrian trajectory prediction. In: AAAI, vol. 36, pp. 542–550 (2022)
10. Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., Lu, J.: Stochastic trajectory prediction via motion indeterminacy diffusion. In: CVPR, pp. 17113–17122 (2022)
11. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: socially acceptable trajectories with generative adversarial networks. In: 2018 IEEE conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018)
12. Halawa, M., Hellwich, O., Bideau, P.: Action-based contrastive learning for trajectory prediction. In: European Conference on Computer Vision, pp. 143–159. Springer (2022)
13. Ho, J., et al.: Imagen video: high definition video generation with diffusion models (2022). arXiv preprint: arXiv:2210.02303
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
16. Huang, J., Gautam, A., Saripalli, S.: Learning pedestrian actions to ensure safe autonomous driving. In: IEEE Intelligent Vehicles Symposium (IV), pp. 1–8 (2023)

17. Ivanovic, B., Pavone, M.: The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: ICCV, pp. 2375–2384 (2019)
18. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Adv. Neural. Inf. Process. Syst. **35**, 26565–26577 (2022)
19. Kesa, O., Styles, O., Sanchez, V.: Multiple object tracking and forecasting: jointly predicting current and future object locations. In: 2022 IEEE Winter Conference on Applications of Computer Vision, pp. 560–569 (2022)
20. Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: a Versatile Diffusion Model for Audio Synthesis (2020). arXiv preprint: arXiv:2009.09761
21. Kosaraju, V., Sadeghian, A., Martín-Martín, R., Reid, I., Rezatofighi, H., Savarese, S.: Social-bigat: multimodal trajectory forecasting using bicycle-gan and graph attention networks. Adv. Neural Inf. Process. Syst. **32**, (2019)
22. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: distant future prediction in dynamic scenes with interacting agents. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 336–345 (2017)
23. Lerner, A., Chrysanthou, Y., Lischinski, D.: Crowds by example. In: Computer Graphics Forum, vol. 26, pp. 655–664. Wiley Online Library (2007)
24. Li, J., Ma, H., Tomizuka, M.: Conditional generative neural system for probabilistic trajectory prediction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6150–6156 (2019)
25. Liang, J., Jiang, L., Niebles, J.C., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: CVPR, pp. 5725–5734 (2019)
26. Mangalam, K., et al.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: ECCV, pp. 759–776. Springer (2020)
27. Mao, W., Xu, C., Zhu, Q., Chen, S., Wang, Y.: Leapfrog diffusion model for stochastic trajectory prediction. In: CVPR, pp. 5517–5526 (2023)
28. Moreno, E., Denny, P., Ward, E., Horgan, J., Eising, C., Jones, E., Glavin, M., Parsi, A.: Mullins: pedestrian crossing intention forecasting at unsignalized intersections using naturalistic trajectories. Sensors **23**(5), 2773 (2023)
29. Robicquet, A., Sadeghian, A., Alahi, A.: Learning social etiquette: human trajectory understanding in crowded scenes. In: ECCV, pp. 549–565. Springer (2016)
30. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
31. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N.: Sophie: an attentive gan for predicting paths compliant to social and physical constraints. In: CVPR, pp. 1349–1358 (2019)
32. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. In: ECCV, pp. 683–700. Springer (2020)
33. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models (2020). arXiv preprint: arXiv:2010.02502
34. Wu, Y., Wang, L., Zhou, S., Duan, J., Hua, G., Tang, W.: Multi-stream representation learning for pedestrian trajectory prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 2875–2882 (2023)
35. Xu, C., et al.: Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In: CVPR, pp. 1410–1420 (2023)
36. Xu, P., Hayet, J.B., Karamouzas, I.: Socialvae: human trajectory prediction using timewise latents. In: ECCV, pp. 511–528 (2022)

37. Yang, R., Srivastava, P., Mandt, S.: Diffusion probabilistic modeling for video generation. Entropy **25**(10), 1469 (2023)
38. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: ECCV. pp. 507–523. Springer (2020)
39. Yuan, Y., Weng, X., Ou, Y., Kitani, K.M.: Agentformer: agent-aware transformers for socio-temporal multi-agent forecasting. In: ICCV, pp. 9813–9823 (2021)
40. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: a unified predictor-corrector framework for fast sampling of diffusion models. Adv. Neural Inf. Process. Syst. **36**, (2024)

RTAT: A Robust Two-Stage Association Tracker for Multi-object Tracking

Song Guo[1](✉), Rujie Liu[1], and Narishige Abe[2]

[1] Fujitsu Research and Development Center Co., Ltd., Beijing, China
{guosong,rjliu}@fujitsu.com
[2] Fujitsu Ltd., Kanagawa, Japan
abe.narishige@fujitsu.com

Abstract. Data association is an essential part in the tracking-by-detection based Multi-Object Tracking (MOT). Most trackers focus on designing a better data association strategy to improve the tracking performance. The rule-based handcrafted association methods are simple and highly efficient but lack generalization capability to deal with complex scenes. While the learnt association methods can learn high-order contextual information to deal with various complex scenes, but they have the limitations of higher complexity and cost. To address these limitations, we propose a Robust Two-stage Association Tracker, named RTAT, where the first-stage association is performed between tracklets and detections to generate tracklets with high purity, and the second-stage association is performed between tracklets to form final trajectories. For the first-stage association, we use a simple data association strategy to generate tracklets with high purity by setting a low threshold for the matching cost in the assignment process. For the second-stage association, we adopt the message-passing GNN framework, which models the tracklet association as a series of edge classification problem in hierarchical graphs, so that it can recursively merge short tracklets into longer ones. Our tracker RTAT ranks first on the test set of MOT17 and MOT20 benchmarks in most of the main MOT metrics: HOTA, IDF1, and AssA. More specifically, RTAT achieve 67.2 HOTA, 84.7 IDF1, and 69.7 AssA on MOT17, and 66.2 HOTA, 82.5 IDF1, and 68.2 AssA on MOT20.

Keywords: Multi-Object Tracking · Data Association · Tracklet Association · Graph Neural Networks · Neural Message Passing

1 Introduction

Multi-Object Tracking (MOT) aims to detect and identify all the objects, and ideally to form one complete trajectory for each object in a video. It is an essential technology for various applications, such as intelligent surveillance, autonomous driving, and robotics. Tracking-by-Detection (TbD) [1, 3, 4, 7] is currently the most effective paradigm for MOT, which contains two steps: object detection and data association. Most trackers [1, 4, 5, 7, 8, 10] focus on designing a better data association strategy to enhance the tracking performance, and various strategies have been proposed which broadly fall into two categories: handcrafted association and learnt association.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 432–447, 2025.
https://doi.org/10.1007/978-3-031-78444-6_28

Matching by cost matrix is usually adopted in handcrafted association methods, where different strategies [1, 4, 5, 7, 8, 10, 33, 34] are designed to match the tracklets and detections based on their distances. Bipartite matching, which is formulated as a Linear Assignment Problem (LAP) and solved by Hungarian algorithm [27], is a commonly used matching strategy. The handcrafted association methods explicitly leverage various cues (e.g., location, motion, appearance, detection scores [1], etc.) to calculate the distances, and design different strategies to construct the cost matrix for identity assignment. Because of their simplicity and efficiency, these methods are very popular in MOT. However, most of them are rule-based, so it is hard and exhausting to design a general association strategy that can deal with various scenes, such as crowded, fast camera motion, night, and low resolution. Another drawback of these methods is that the association error cannot be fixed once it occurs.

In the learnt association methods, the data association is usually done implicitly based on the learnt association feature through a neural network, such as Transformer [28–30], Graph Neural Networks (GNN) [17, 18, 20–22]. These methods learn to extract high-order association feature from multiple sources of information (e.g., spatial and temporal, appearance, motion, etc.) in a data-driven manner. In the transformer-based methods [28, 30], data association is implemented by using query propagation, where either parallel or sequential interactions between the (tracking and/or detection) query and image feature are utilized. However, the training strategies are highly complex. Furthermore, a large amount of data is required to train the Transformer models, which can't be easily satisfied due to the limited scales of MOT datasets [25, 26]. In Graph-based methods [18, 21, 22], the detections and their connections are respectively represented as nodes and edges in a graph, and the data association is solved as an edge classification problem. However, the size of the graph is proportional to video duration and object quantity in the video, therefore, large graphs will be constructed for long videos or crowded scenes, leading to high computational complexity and large memory consumption [21]. Generally speaking, the learnt association methods can leverage high-order information to deal with more complex scenes, but they have the limitations of higher complexity and cost.

In order to effectively utilize the advantages and address the limitations of these two kinds of data association methods, we propose a Robust Two-stage Association Tracker, named RTAT. The first-stage association is performed between tracklets and detections to generate tracklets with high purity, and the second-stage association is performed between tracklets to form complete trajectories.

In the first-stage association, we use a simple data association strategy to generate tracklets with high purity. This is done by setting a low threshold for the matching cost in the identity assignment process. The generated tracklets have higher purity and less identity switches. As a result, the number of tracklets will increase, and the fragmentations problem will be solved in the second-stage by using tracklets association.

In the second-stage association, we merge the tracklets into complete trajectories by using GNN. Our method models the tracklet merging as a series of edge classification problem in hierarchical graphs, which can recursively merge short tracklets into longer ones and finally form complete trajectories. We use the message passing mechanism [21, 31] to update the graphs and learn features for nodes and edges, and then perform edge classification based on the final edge feature. This process is hierarchically performed on graph in each level. Since the number of tracklets is much smaller than that of detections, our GNN model can take all the tracklets in a video sequence as input and it can effectively deal with the problem of higher computational complexity and memory consumption in existing graph-based tracking methods. Experiments on two of the most popular MOT benchmarks: MOT17 [25] and MOT20 [26], demonstrate the effectiveness of our method.

2 Related Works

2.1 Handcrafted Association

The handcrafted association methods match the detections to the tracklets based on well-designed cost matrix by leveraging various strategies [1, 4, 5, 7, 8, 10, 33, 34]. Intersection over union (IoU) and appearance distance are the most commonly used metrics to construct the cost matrix. Motion model is adopted to predict the locations of tracklets to calculate the IoU distance with the detections, while person Re-identification (ReID) model is used to extract the appearance features to calculate the appearance distance. Generally, IoU distance is more useful in short-term matching, while appearance information is more accurate in long-term matching.

There are four main research directions in the handcrafted association methods. (1) Learn more accurate motion models: Kalman filter (KF) and its variants [1, 4, 5, 32–34], camera motion compensation (CMC) [4, 5], etc. (2) Extract more discriminative ReID feature: independent ReID model [4, 7, 34], occlusion-aware ReID feature [35], dynamic ReID feature [4, 5], etc. (3) Design more sophisticated strategy to construct the cost matrix: different combination of the IoU and appearance distance, such as weighted sum [5, 7, 36], minimum cost [4], etc. (4) Develop better matching strategy: single matching [33], cascade matching [2, 3, 34], etc. Many researchers have invested a great deal of time and effort in designing a better data association strategy. However, it is hard and exhausting to design a generic data association that can deal with various scenes. Therefore, we turn to use a simple association method to obtain tracklets with high purity, and further merge them by using tracklet association.

2.2 Graph-Based Association

Graph-based methods perform data association on constructed graphs, where nodes represent detections and edges indicate linkage between them. The data association is formulated as a graph optimization problem, which is solved by different algorithms, such as network flows [15], k-shortest paths [16], minimum cost lifted multicut [17], lifted disjoint paths [18, 19], etc. Recently, GNN [20] is introduced as an extension of neural networks that can operate on graph. GNN can extract high-order contextual information by adopting a message passing mechanism, which propagates the information encoded in the features of neighboring nodes and edges across the graph [20–23]. MPNTrack [21] designs a tracker based on Message Passing Network to learn features for nodes and edges and treats the data association as an edge classification task. SUSHI

[22] proposes a unified tracker for short and long-term tracking by using a hierarchy of message passing GNNs. SGT [23] employs GNNs to recover the missed detections to enhance the tracking performance for online graph tracker.

In contrast to handcrafted association, graph-based association methods seek for global optimization over longer range frames. Specially, GNN-based methods can learn high-order information through message passing, and therefore they can achieve better tracking performance [21, 22]. However, it needs to construct very large graphs for long videos or videos in crowded scenes, which brings the issues of higher complexity and cost [21]. In our work, we build graph for tracklet association, where the scale of the graph is much smaller. Therefore, it can effectively solve the above-mentioned problems and still utilize the advantages of graph-based association.

2.3 Tracklet Association

Tracklet association [11–13] has drawn much attention in TbD based MOT. Several methods [11] exploit the idea of multi-level association, which first generates short tracklets in adjacent frames and then merges them into trajectories by tracklet association. Some works [12, 13] follow the split-merge pipeline to refine the tracking results of existing trackers, and tracklet association is employed in the merging process. TAT [11] employs a Multi-Layer Perceptron (MLP) to link detections in adjacent frames to generate short tracklets, and then trains a network flow to associate the tracklets into trajectories. ReMOT [12] splits tracklets by using appearance and motion features, and then associates the tracklets by hierarchical clustering on a designed distance matrix. [13] proposes a tracklet booster for existing trackers, which trains a Splitter to split tracklets into small pieces, and then learns a Connector to merge the tracklet pieces that are from the same identity. These methods generate short tracklets either in a sliding window with limited size or by splitting existing tracklets into small pieces. The generated tracklets are often too short, which will increase the burden for the following tracklet association. Furthermore, performing tracklet association by using the message-passing GNN has not been fully exploited in these methods.

3 Methodology

3.1 Motivation

The motivation of our Robust Two-stage Association Tracker (RTAT) is simple and effective. It is hard and exhausting to design a generic data association strategy that can handle various scenes by explicitly leveraging simple cues, while learnt association methods have the limitations of higher complexity and cost, although they can learn high-order information to deal with more complex scenes. Therefore, we propose to use simple cues to generate clean tracklet pieces, and then employ GNN for tracklet association to obtain the final trajectories. RTAT consists of two-stage associations, where the first-stage association is performed between tracklets and detections to generate tracklets with high purity, and the second-stage association is performed between tracklets to obtain complete trajectories. The workflow of RTAT is shown in Fig. 1. We will describe the details of our method in the following sections.

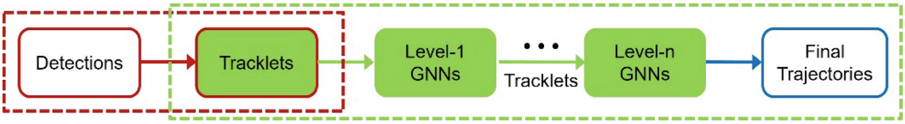


Fig. 1. The workflow of our robust two-stage association tracker (RTAT). The first-stage association (red dashed box) generate tracklets with high purity from detections, and the second-stage association (green dashed box) merge short tracklets into longer ones by using hierarchical GNNs and finally form complete trajectories (Color figure online)

3.2 Method Formulation

Given a video sequence with K frames and a set of detections $D = \{d_i, i \in [1, M]\}$, where M is the total number of detections obtained from the K frames. Each detection d_i can be represented by its bounding box coordinates, image region, and timestamp. Let us define the set of tracklets as $T = \{t_j, j \in [1, N]\}$, where N is the number of tracklets in the video sequence. Each tracklet consists of a set of detections $t_j = \left\{d_j^i, i \in [1, n_j]\right\}$, where n_j is the number of detections in the tracklet of t_j. The aim of our first-stage association is to generate the initial set of tracklets T.

In the task of tracklet association, we construct an undirected graph $G = (V, E)$, where nodes represent the tracklets (e.g., $V = T$) and edges indicate the connections between them. The set of edges can be denoted as $E = \{e_{ij} = (t_i, t_j) \in N \times N, i \neq j\}$, where e_{ij} represents the linkage of a pair of tracklets (t_i, t_j). We introduce a binary variable $y_{e_{ij}}$ to indicate whether t_i and t_j are from the same identity. Specifically,

$$y_{e_{ij}} = \begin{cases} 1, \exists I_t \in I, s.t.(t_i, t_j) \in I_t \\ \quad 0, otherwise \end{cases} \tag{1}$$

where I is the set of identities in a given video sequence. An edge is active if its value $y_{e_{ij}} = 1$, otherwise, it is inactive. We perform edge classification to predict the values of each edge based on the learnt edge feature and merge the tracklets that belong to the same identity, i.e., nodes are linked by active edge. Different from other graph-based association methods which take detections in a short video clip with limited number of frames as input, we take all the tracklets in a video as input to obtain the final trajectories.

3.3 First-Stage: Tracklet Generation

The aim of the first-stage association is to generate tracklets with high purity. Any tracker can be used in this stage, but we prefer trackers with simple data association strategy, such as ByteTrack [1], BoT-SORT [4]. The matching is usually done by bipartite matching, which is solved by Hungarian algorithm [27]. In the assignment process, we set a cost threshold th_c for possible matching and reject the matchings with higher cost than th_c. For simplification, we normalize the value of the cost in cost matrix to be [0, 1] for different tracker. By setting a lower cost threshold th_c, we can obtain tracklets with higher purity. Consequently, there are less identity switches in each tracklet, but the number of tracklet fragments will increase. We will focus on solving the fragmentation problem in the next stage by using tracklets association.

3.4 Second-Stage: Tracklet Association

The aim of the second-stage association is to merge the tracklet pieces into trajectories. We perform the tracklet association based on the framework of message-passing GNN [21, 22, 31]. An illustration of the tracklets merging process is shown in Fig. 2.

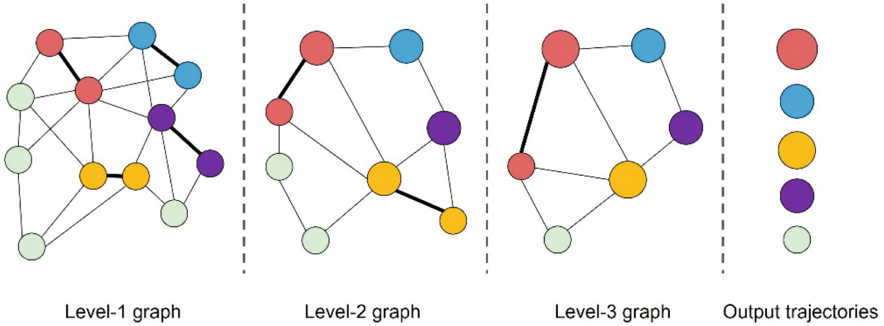


Fig. 2. An illustration of the tracklets merging process in hierarchical graphs. The bold edges are classified as active, and the nodes linked by active edge will be merged in current level. The final trajectories are listed in the last column.

Our method models the tracklet merging as a series of edge classification problem in hierarchical graphs, which can recursively merge short tracklets into longer ones. We use the message passing mechanism to update the feature vectors for nodes and edges across the graph and the edge classification is performed based on the final edge feature. This process is performed hierarchically for graph in each level and the workflow of each level contains four main steps:

Graph Construction. We construct an undirected graph $G = (V, E)$, where each node represents a tracklet and each edge indicates the possible connection for a pair of tracklets. Compared to detection association, the number of nodes is largely reduced for tracklet association. However, the number of edges is still very large if all the connections between each pair of nodes are considered. Moreover, it will cause a severe label imbalance between active and inactive edges, which can deteriorate the performance of the tracklet association. Therefore, we only consider the edge between a pair of nodes that have no temporal overlap. We further limit the number of edges for each node to be K, which are selected by its top K nearest neighbors according to the similarity measures of appearance, motion, and spatial position. As a result, we construct a sparse graph with limited number of edges, which can reduce the computational complexity, and alleviate the problem of edge label imbalance.

Graph Initialization. The node feature vector is initialized by the feature of its corresponding tracklet. We first extract appearance features for all the detections in each tracklet by feeding their image patches into a pretrained Convolutional Neural Network (CNN), and then calculate their average feature as the tracklet feature. The averaged tracklet feature is more robust to motion blur, partial occlusion, and illumination change than single detection appearance feature. The tracklet feature is fed into a node encoder E_n^{enc}, whose output is used to initialize its corresponding node feature.

The edge feature vector is initialized with the output of an MLP, the input of which is a concatenated vector of the association features from two connected tracklets. We adopt spatial and temporal distance, appearance and motion information to construct the initial feature vector, which is an extension of MPNTrack [21] and SUSHI [22].

For a pair of tracklets T_a and T_b with their detection box coordinates and timestamps, which can be described as $T_a = \{(x_i, y_i, w_i, h_i, t_i), i \in [a_1, a_n]\}$ and $T_b = \{(x_j, y_j, w_j, h_j, t_j), j \in [b_1, b_n]\}$, where $[a_1, a_n]$ and $[b_1, b_n]$ are the frame range of T_a and T_b respectively. Assuming that T_a ends before T_b starts, so we have $t_{a_n} < t_{b_1}$. We use their closest detection boxes to compute the relative spatial distance and scale difference, which is formulated as:

$$\left[\frac{2(x_j - x_i)}{h_j + h_i}, \frac{2(y_j - y_i)}{h_j + h_i}, log\frac{h_j}{h_i}, log\frac{w_j}{w_i}\right] \tag{2}$$

where $i = a_n$ and $j = b_1$. Supposing the FPS (Frames Per Second) of the given video is *fps*, we calculate their time difference by the following equation:

$$\left(t_{b_1} - t_{a_n}\right)/fps \tag{3}$$

To encode the appearance information, we use the Euclidean distance of the tracklets feature and the average cosine similarity of the top L closest detections for each pair of tracklets, which can be formulated as:

$$\left[\|app_{T_b}^{avg} - app_{T_a}^{avg}\|_2, \frac{1}{L * L}\sum_{i=1}^{L}\sum_{j=1}^{L} cos(app_{T_a}^{i}, app_{T_b}^{j})\right] \tag{4}$$

where the first distance encodes global appearance discrepancy between two tracklets, and the second similarity describes local appearance similarity, which is helpful to remove the influence of large appearance variations inside a tracklet, such as large pose, long-time occlusion, and etc.

The tracklets belong to the same trajectory are expected to satisfy motion consistency [22], so we add the motion information into the edge feature. We employ Kalman Filter (KF) to model the object's motion and predict its position in a desired frame for each tracklet. For a pair of tracklets T_a and T_b, we calculate their middle frame $t_{mid} = t_{a_n} + \left(t_{b_1} - t_{a_n}\right)/2$, and predict their box positions at this frame which are respectively denoted as $pred\_box_{T_a}^{t_{mid}}$ and $pred\_box_{T_b}^{t_{mid}}$. We adopt the Generalized Intersection over Union (GIOU) [6] score of these two estimated boxes to measure their motion consistency, which is formulate as:

$$GIOU\,(pred\_box_{T_a}^{t_{mid}}, pred\_box_{T_b}^{t_{mid}}) \tag{5}$$

Finally, the concatenation of the feature vectors from Eq. 2 to Eq. 5 is fed into an edge encoder E_e^{enc} to obtain the initial edge feature. Both the node coder E_n^{enc} and edge encoder E_e^{enc} are light-weight MLP networks.

Graph Update. We employ the message-passing mechanism to update the features for nodes and edges [21, 22, 31]. During each step of message-passing, every node and edge aggregates their received information, and then combine the incoming information with their own to update their feature vectors [31]. Specifically, for the construct graph $G = (V, E)$, we obtain the initial feature vector f_i^0 and $f_{(i,j)}^0$ for each node $i \in V$ and each edge $(i,j) \in E$ from the graph initialization step. The mechanism of message-passing is to propagate messages between neighboring nodes and edges across the graph. The propagation is performed by alternately updating the features of edges and nodes, which is divided into two steps: update edge feature using neighboring nodes and update node feature using neighboring edges. Both updates are sequentially performed for L iterations. For each iteration $l \in [1, L]$, the edges and nodes features are updated as follows:

$$f_{(i,j)}^l = U_e\left(\left[f_i^{l-1}, f_j^{l-1}, f_{(i,j)}^{l-1}\right]\right) \tag{6}$$

$$m_{(i,j)}^l = U_n\left(\left[f_i^{l-1}, f_{(i,j)}^l\right]\right) \tag{7}$$

$$f_i^l = \phi\left(\left\{m_{(i,j)}^l\right\}_{j \in N_i}\right) \tag{8}$$

where U_e and U_n are learnable networks (e.g., MLP) that aggregate information from neighboring nodes and edges. N_i is the set of nodes adjacent to node i, and ϕ denotes an order-invariant operation, e.g., maximum, summation or average. After L iterations, we obtain the final node and edge features, which contain high-order contextual information from neighboring nodes and edges in a distance of L along the graph.

Edge Classification. We use an MLP with sigmoid function as the edge classifier C_e^{class} and then perform edges classification based on their final features $f_{(i,j)}^L$, which is formulated as:

$$y_{(i,j)} = C_e^{class}\left(f_{(i,j)}^L\right), (i,j) \in E \tag{9}$$

where the predicted edge score $y_{(i,j)} \in (0, 1)$. The scores are further rounded to binary values using the exact rounding solution described in [21]. The edges are classified as active or inactive, and the tracklets linked by the active edges are merged into longer ones.

Afterwards, we update the set of tracklets and hierarchically perform these four steps to obtain the final trajectories, i.e., graph construction, graph initialization, graph update and edge classification.

Data Augmentation. In the tracklet association stage, a training sample consists of a video sequence and a set of tracklets. There are very few training samples in MOT17 [25] and MOT20 [26], which are 7 and 4 respectively. Therefore, we introduce data augmentations from both video-level and tracklet-level to train the GNN networks with higher robustness and generality. In video-level augmentation, we generate more video clips from the original video sequences. We sample a video clip in every 50 frames (i.e., start points), and the start frame is randomly selected with a fluctuation of 15 frames at each start point. The length of a video clip is randomly selected from 25% to 100% of the length for the whole video. In tracklet-level augmentation, we generate more sets of tracklets by adopting different data association strategies under different cost thresholds in Sect. 3.3.

Training GNN. We use the same GNN architectures for graphs in different hierarchical levels. Since the aims of all hierarchical levels are the same, which is to merge tracklets that belong to the same identity into longer ones, we also share the parameters of GNNs for all hierarchical levels. The difference among different levels is the lengths and numbers of tracklets, so we design a level adapter, which is a learnable vector that has the same dimension with the edge features. The level adapter is then added to the edge feature for each level, and it will help the GNN model to learn the most important cues for each level in a data-driven manner. We adopt the focal loss to train the edge classifier, and the final loss is a summation of losses in all levels.

4 Experiments

4.1 Experimental Settings

Datasets. We conduct our experiments on two of the most popular MOT benchmarks: MOT17 [25] and MOT20 [26], under the "private detection" protocol. MOT17 [25] contains 14 video sequences which are filmed under a variety of conditions, such as camera motions (moving, static), viewpoints (high, medium, low) and weather conditions (night, sunny, cloudy, indoor, etc.). MOT20 [26] contains 8 video sequences in very crowded scenes. Both datasets are split into training and testing sets.

Metrics. Our method focuses on robust data association, so we adopt HOTA [24] and IDF1 [39] as the main metrics. We also use the metrics MOTA, AssA [24], and IDs to provide comparisons from more perspectives. HOTA maintains a good balance between the accuracy of object detection and association. IDF1 measures the identity preservation ability and focus more on the association ability. AssA is used to evaluate the association performance, while MOTA focuses on the detection performance. Moreover, we adopt the number of tracklets as a metric to measure how many tracklets are there in a video after data association.

We introduce a new metric, named High Purity Rate (HPR), to measure the rate of high purity tracklets in all the tracklets for a given video. We use the definition of mostly tracked (MT) as reference to define high purity. A tracklet has high purity if more than 80% of its detections are from the same identity.

Implementation Details. We train a YoloX detector to obtain detections for both MOT17 and MOT20 following [1]. We adopt three popular trackers, i.e., ByteTrack [1], BoT-SORT [4], and Deep OC-SORT [7], to generate short tracklets in the first-stage association. The following part describes the implementation details of tracklet association in RTAT. We train a ReID model using ResNet50 following [21] to extract appearance feature. After the convolutional layers in ResNet50, a node encoder is added to reduce the dimension of node feature to 32. All of the networks are light-weight MLPs, and their detailed architectures are shown in Fig. 3.

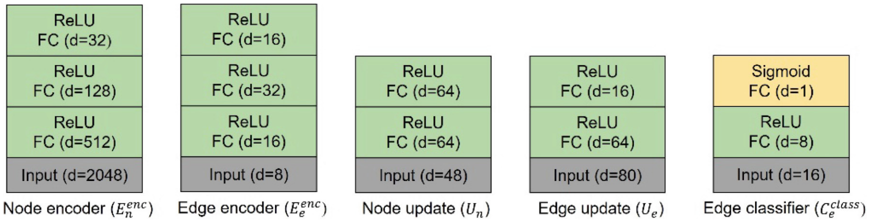


Fig. 3. Detailed architectures for all networks. d means the output dimensions for each layer.

We use all the tracklets in a video sequence and perform data augmentation from both video-level and tracklet-level to train the GNN networks, as explained in Sect. 3.4. We use three levels of hierarchical graphs for tracklet association. GNNs in all the three levels are jointly trained for 500 epochs using the Adam optimizer with a learning rate of 3×10^{-4} and a weight decay of 10^{-4}. We set $\gamma = 1$ for focal loss, $K = 10$ to limit the number of edges connected to each node, and $L = 12$ for the steps of message-passing in all GNNs. We further apply linear interpolation to fix the missing detections in the final trajectories during inference.

4.2 Ablation Studies

We perform 3-fold cross-validation on the MOT17 training set for ablation studies following the experimental setup in [21] and use IDF1 as the primary metric. We study three main aspects of our method in this Sect. 1) How to select the proper cost threshold th_c to obtain tracklets with high purity. 2) How different training strategies affect the tracklet-association performance. 3) The effect of using different data association methods to generate tracklets in the first-stage.

Obtain Tracklets with High Purity. We select BoT-SORT-ReID [4] to generate tracklets for analysis in this experiment. We set a cost threshold th_c to reject the matchings that have higher cost in the assignment process. Obviously, lower cost threshold can obtain more tracklets with higher purity. However, it is meaningless if we set a very small threshold, such as 0, where each detection is a tracklet with 100% purity. We need to keep a balance between the number of tracklets and the high purity rate (HPR). We list the tracking performance in the first-stage and second-stage with different cost threshold in Table 1. The number of tracklets in the ground-truth is also listed for comparison.

In the first-stage association, the number of tracklets and HPR constantly increase as we decrease the cost threshold, while the IDF1 score decreases slightly when $th_c \geq 0.2$. We perform tracklet association based on the tracklets generated in the first-stage, the IDF1 score has increased after the second-stage association under all cost thresholds, even when $th_c = 0.7$ which is the default setting in BoT-SORT [4]. The best result is achieved when $th_c = 0.2$, which has the highest IDF1 score and the fewest ID switches. When $th_c = 0.1$, the largest increasement on IDF1 (i.e., 19.2%) occurs, however its IDF1 score is lower than that of $th_c = 0.2$ in both stages, and it has much more tracklet (4964 versus 1693) which will bring more computational cost during inference. Therefore, we choose 0.2 as the cost threshold in our experiments.

Table 1. The tracking performance in the first-stage and second-stage with different cost threshold. The number of tracklets in ground-truth is also listed for comparison

First-stage association				Second-stage association			GT
th_c	IDF1 ↑	#Tracklets	HPR	IDF1 ↑	#Tracklets	IDs ↓	#Tracklets
0.7	85.0	753	84.5	86.0	641	291	546
0.5	85.2	979	87.3	86.5	630	282	
0.4	84.5	1,091	89.0	87.6	623	278	
0.3	84.3	1,218	91.1	88.2	615	265	
0.2	83.6	1,693	93.4	**88.5**	**612**	**261**	
0.1	67.1	4,964	98.1	86.3	621	276	

Table 2. The performance of tracklet association with different hierarchical levels

# *HL*	IDF1 ↑	IDs ↓	# Tracklets
0	83.6	1,344	1,693
1	86.2	406	1,021
2	87.6	287	728
3	**88.5**	**261**	**612**
4	88.6	258	608
5	88.6	257	607

The Effect of Different Training Strategies. We adopt BoT-SORT-ReID with a cost threshold $th_c = 0.2$ to generate tracklets in the first-stage association.

Firstly, we evaluate how the number of hierarchical levels (*HL*) in the tracklet association effect the metrics of IDF1, IDs and the number of tracklets. The number of *HL* varies from 0 to 5, and their tracking metrics are listed in Table 2. $HL = 0$ means the performance for the tracklets obtained in the first-stage. With the increase of *HL*, the IDF1 constantly increase, the ID switches and the number of tracklets constantly decrease. At the same time, both the increase and the decrease become smaller and smaller. The increasement of IDF1 can be ignored when the number of *HL* is bigger than 3. Furthermore, larger number of hierarchical levels will increase the time and memory costs for the tracklet association. Hence, we set $HL = 3$ in our experiments.

Secondly, we evaluate the effect of using different data augmentation strategies in training GNN networks. We take all the tracklets in a video sequence to build graph and train GNNs, however, there are only 7 and 4 training samples in MOT17 and MOT20 respectively, which are very few to learn GNNs with higher robustness and generality. As described in Sect. 3.4, we design data augmentations from both video-level and tracklet-level to generate more training samples. The results of using different combinations of data augmentations are listed in Table 3. We can see that both the data augmentation strategies can improve the IDF1 score separately, and their combination can achieve higher improvement. The results demonstrate the effectiveness of our data augmentation methods in training robust GNN networks.

Table 3. The performance of tracklet association using different data augmentation

Data Augmentation		Tracking Metrics	
Video-level	Tracklet-level	IDF1 ↑	IDs ↓
		85.5	293
√		86.7	278
	√	87.2	272
√	√	**88.5**	**261**

The Effect of Using Different Data Association Methods in the First-Stage. We use three popular trackers, i.e., ByteTrack [1], BoT-SORT [4], Deep OC-SORT [7], for the first-stage association, and compare their performance before and after the tracklet association in second-stage. The results are listed in Table 4. There are big differences among the three trackers on all the three metrics in the first-stage association, however, the differences are largely reduced after the second-stage association. We can see that our method can obtain very similar tracking performance no matter which tracker is used, which indicates that simple data association strategy is good enough for the first-stage.

Table 4. The performance of using different data association methods in the first-stage.

First-stage association				Second-stage association		
Tracker	IDF1 ↑	#Tracklets	HPR	IDF1 ↑	#Tracklets	IDs ↓
ByteTrack	76.6	1,571	91.9	88.0	623	276
BoT-SORT	83.6	1,693	93.4	**88.5**	**612**	**261**
Deep OC-SORT	81.2	1,264	89.7	88.2	617	264

4.3 Benchmarks Evaluation

We present the results of the state-of-the-art trackers on the test set of MOT17 and MOT20 benchmarks under the "private detection" protocol in Tables 5 and 6, respectively. All the results are obtained from the official MOTChallenge server [37].

We adopt ByteTrack and BoT-SORT to generate tracklets in the first-stage association, which are named RTAT-ByteTrack and RTAT-BoT-SORT, respectively. Both versions of our method outperform all the other trackers in almost all the main metrics.

Table 5. Comparison of the state-of-the-art methods under the "private detection" protocol on MOT17 test set. The trackers are sorted by HOTA. The best results are shown in **bold**.

Tracker	HOTA ↑	IDF1 ↑	MOTA ↑	AssA ↑	IDs ↓
ByteTrack [1]	63.1	77.3	80.3	62.0	2,196
StrongSORT [5]	64.4	79.5	79.6	64.4	1,194
Deep OC-SORT [7]	64.9	80.6	79.4	65.9	1,023
BoT-SORT [4]	65.0	80.2	80.5	65.5	1,212
MotionTrack [8]	65.1	80.1	81.1	65.1	1,140
ConfTrack [36]	65.4	81.2	80.0	66.3	1,155
CBIOU [14]	66.0	82.5	**82.8**	66.1	1,194
PIA [38]	66.0	81.1	82.2	65.8	1,026
ImprAsso [10]	66.4	82.1	82.2	66.6	924
SUSHI [22]	66.5	83.1	81.1	67.8	1,149
RTAT-ByteTrack (ours)	67.0	84.4	80.1	69.3	942
RTAT-BoT-SORT (ours)	**67.2**	**84.7**	80.4	**69.7**	**912**

Table 6. Comparison of the state-of-the-art methods under the "private detection" protocol on MOT20 test set. The trackers are sorted by HOTA. The best results are shown in **bold**.

Tracker	HOTA ↑	IDF1 ↑	MOTA ↑	AssA ↑	IDs ↓
ByteTrack [1]	61.3	75.2	77.8	59.6	1,223
StrongSORT [5]	62.6	77.0	73.8	64.0	770
MotionTrack [8]	62.8	76.5	78.0	61.8	1,165
BoT-SORT [4]	63.3	77.5	77.8	62.9	1,313
FineTrack [9]	63.6	79.0	77.9	63.8	980
Deep OC-SORT [7]	63.9	79.2	75.6	65.7	779
SUSHI [22]	64.3	79.8	74.3	67.5	706
ImprAsso [10]	64.6	78.8	**78.6**	64.6	992
PIA [38]	64.7	79.0	78.5	64.9	1,023
ConfTrack [36]	64.8	80.2	77.2	66.2	**702**
RTAT-ByteTrack (ours)	65.9	82.1	78.1	67.7	817
RTAT-BoT-SORT (ours)	**66.2**	**82.5**	78.4	**68.2**	787

Our method can achieve the best performance in all association related metrics, i.e., HOTA, IDF1, and AssA, on both benchmarks, which demonstrate the effectiveness of our method for data association. For example, RTAT-BoT-SORT outperforms the tracker in second place by a large margin (i.e., +1.4 HOTA, +2.3 IDF1, and +1.9 AssA) on MOT20 benchmark.

Both RTAT-ByteTrack and RTAT-BoT-SORT outperform their respective baseline by a large margin on both MOT17 and MOT20. It is worth noting that RTAT-ByteTrack can achieve similar performance with RTAT-BoT-SORT in all metrics. The performance gap between ByteTrack and BoT-SORT are filled by using the tracklet associations in our method. This observation demonstrates that simple association strategy is enough to generate tracklets with high purity for the tracklet association in the second-stage, and there is no need to design more sophisticated data association strategy by investing a great deal of time and effort.

5 Conclusion

We propose a Robust Two-stage Association Tracker (RTAT), which can achieve higher association performance by utilizing the advantages of two kinds of data association methods: the simplicity and efficiency of handcrafted association methods and the effective high-order contextual information of learnt association methods. We use a simple data association method to generate tracklets with high purity in the first-stage and use message-passing GNNs to perform tracklet association in the second-stage. We further design data augmentation strategies from video-level and tracklet-level to improve the generalization ability of our tracklet association model. Ablation studies and MOT benchmarks results validate the effectiveness of our method. We hope our work is helpful to release researchers from the hard and exhausting work of designing more and more sophisticated data association strategy to obtain minor improvement in tracking performance. We also expect this work can push forward the development of multiple-object tracking.

References

1. Zhang, Y., Sun, P., Jiang, Y., et al.: Bytetrack: multi-object tracking by associating every detection box. In: European Conference on Computer Vision, pp. 1–21. Springer Nature Switzerland, Cham (2022)
2. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2018)
3. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9686–9696 (2023)
4. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: BoT-SORT: Robust Associations Multi-pedestrian Tracking (2022). arXiv preprint, arXiv:2206.14651
5. Du, Y., Zhao, Z., Song, Y., et al.: Strongsort: make deepsort great again. IEEE Trans. Multimed. **25**, 8725–8737 (2023)
6. Rezatofighi, H., Tsoi, N., Gwak, J., et al.: Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)

7. Maggiolino, G., Ahmad, A., Cao, J., Kitani, K.: Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 3025–3029. IEEE (2023)
8. Qin, Z., Zhou, S., Wang, L., et al.: Motiontrack: learning robust short-term and long-term motions for multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17939–17948 (2023)
9. Ren, H., Han, S., Ding, H., et al.: Focus on details: online multi-object tracking with diverse fine-grained representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11289–11298 (2023)
10. Stadler, D., Beyerer, J.: An improved association pipeline for multi-person tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3170–3179 (2023)
11. Shen, H., Huang, L., Huang, C., Xu, W.: Tracklet Association Tracker: An End-to-End Learning-Based Association Approach for Multi-object Tracking (2018). arXiv preprint: arXiv:1808.01562
12. Yang, F., Chang, X., Sakti, S., Wu, Y., Nakamura, S.: ReMOT: a model-agnostic refinement for multiple object tracking. Image Vis. Comput. **106**, 104091 (2021)
13. Wang, G., Wang, Y., Gu, R., et al.: Split and connect: a universal tracklet booster for multi-object tracking. IEEE Trans. Multimed. **25**, 1256–1268 (2022)
14. Yang, F., Odashima, S., et al.: Hard to track objects with irregular motions and similar appearances? Make it easier by buffering the matching space. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4799–4808 (2023)
15. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
16. Berclaz, J., Fleuret, F., et al.: Multiple object tracking using k-shortest paths optimization. IEEE Trans. Pattern Anal. Mach. Intell. **33**(9), 1806–1819 (2011)
17. Tang, S., Andriluka, M., Andres, B., Schiele, B.: Multiple people tracking by lifted multicut and person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3539–3548 (2017)
18. Hornakova, A., Henschel, R, et al: Lifted disjoint paths with application in multiple object tracking. In: International Conference on Machine Learning, pp. 4364–4375. PMLR (2020)
19. Hornakova, A., Kaiser, T., Swoboda, P., et al.: Making higher order mot scalable: an efficient approximate solver for lifted disjoint paths. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6330–6340 (2021)
20. Scarselli, F., Gori, M., Tsoi, A.C., et al.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
21. Brasó G., Leal-Taixé L.: Learning a neural solver for multiple object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6247–6257 (2022)
22. Cetintas, O., Brasó, G., and Leal-Taixé, L.: Unifying short and long-term tracking with graph hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22877–22887 (2023)
23. Hyun, J., Kang, M., Wee, D., Yeung, D.Y.: Detection recovery in online multi-object tracking with sparse graph tracker. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4850–4859 (2023)
24. Luiten, J., Osep, A., Dendorfer, P., et al.: Hota: a higher order metric for evaluating multi-object tracking. Int. J. Comput. Vis. **129**, 548–578 (2021)
25. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: a Benchmark for Multi-object Tracking (2016). arXiv preprint, arXiv:1603.00831

26. Dendorfer, P., Rezatofighi, H., Milan, A., et al.: Mot20: a Benchmark for Multi Object Tracking in Crowded Scenes (2020). arXiv preprint, arXiv:2003.09003
27. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**(1–2), 83–97 (1955)
28. Zeng, F., Dong, B., Zhang, Y., et al.: Motr: end-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision, pp. 659–675. Springer Nature, Cham (2022)
29. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: multi-object tracking with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8844–8854 (2022)
30. Zhang, Y., Wang, T., Zhang, X.: Motrv2: bootstrapping end-to-end multi-object tracking by pretrained object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22056–22065 (2023)
31. Gilmer, J., Schoenholz, S. S., Riley, P. F., et al.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning, pp. 1263–1272. PMLR (2017)
32. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Fluids Eng. **82**(1), 35–45 (1960)
33. Bewley, A., Ge, Z., Ott, L., et al.: Simple online and realtime tracking. In 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464–3468. IEEE (2016)
34. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649. IEEE (2017)
35. Stadler, D., Beyerer, J.: Improving multiple pedestrian tracking by track management and occlusion handling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10958–10967 (2021)
36. Jung, H., Kang, S., Kim, T., Kim, H.: ConfTrack: Kalman filter-based multi-person tracking by utilizing confidence score of detection box. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 6583–6592 (2024)
37. MOTChallenge Homepage. https://motchallenge.net/
38. Stadler, D., Beyerer, J.: Past information aggregation for multi-person tracking. In: 2023 IEEE International Conference on Image Processing (ICIP), pp. 321–325. IEEE (2023)
39. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision, pp. 17–35. Springer International Publishing, Cham (2016)

Inverse DLT Method for One-Sided Radial Distortion Homography

Gaku Nakano(✉)

NEC Corporation, Kawasaki, Japan
g-nakano@nec.com

Abstract. In this paper, we present a novel linear method for the simultaneous estimation of the homography matrix and one-sided radial lens distortion. Initially, we highlight that Fitzgibbon's method, commonly recognized as a DLT method for this problem, is inadequate for handling noisy data. Subsequently, we formulate a new DLT method incorporating lens distortion by considering the inverse homography transformation. The proposed method, termed invDLT, provides two solutions: the minimal case with 4.5 point pairs and the least-squares case with more than five point pairs. We conduct extensive experiments on both synthetic and real image data, revealing that invDLT substantially outperforms conventional methods in terms of estimation accuracy, robustness to outliers, and computational efficiency.

Keywords: Homography matrix · Radial lens distortion · Direct linear transform

1 Introduction

The planar homography is a geometric transformation of a plane between two different views. The homography transformation is parameterized as a 3 × 3 matrix, and estimating the homography matrix from images is one of the most fundamental procedures for computer vision applications such as camera calibration [21], augmented reality (AR) [19], and visual odometry [9].

The homography matrix can be calculated using at least four point correspondences, and the direct linear transform (DLT) method has been known as the most standard solution [10]. The 4-point DLT method assumes no lens distortion, so the estimation accuracy degrades for images taken with a camera equipped with a wide-angle lens.

To deal with this issue, various approaches have been investigated for joint estimation of the homography matrix and lens distortion [4,8,13]. Fitzgibbon [8] proposed a novel lens distortion model called the division model and showed the first DLT-based solution that finds the homography matrix and lens distortion using five point pairs. Fitzgibbon assumes that the cameras used to capture two images are identical, i.e., the cameras share the same lens distortion. For the case when two different cameras take images, Kukelova et al. [13] utilized a

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 448–462, 2025.
https://doi.org/10.1007/978-3-031-78444-6_29

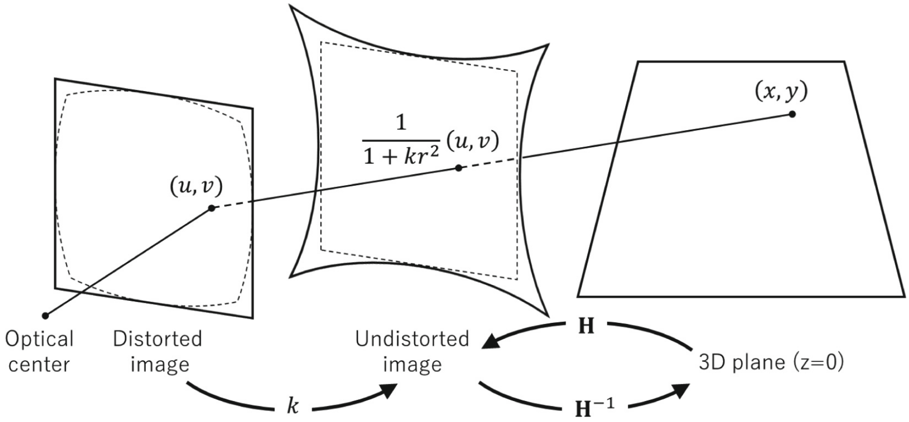


Fig. 1. One-sided radially distorted homography transformation. While Fitzgibbon's method finds the forward homography $\mathtt{H}$ and the backward distortion k, the proposed invDLT estimates both in the backward direction, i.e., the inverse homography $\mathtt{H}^{-1}$ and distortion k.

Grobner basis technique and developed a 6-point minimal method, which simultaneously estimates the homography matrix and two different lens distortions. These methods assume that a 3D plane is captured by a single camera (or two cameras) with a wide-angle lens, i.e., both images are distorted. However, in applications such as camera calibration and AR, one image is a chessboard or a fiducial marker unaffected by lens distortion. Therefore, the methods above that assume lens distortion in both images are inappropriate for such situations of one-sided radial distortion homography.

Another approach for finding the homography with one-sided radial distortion is to solve the uncalibrated perspective-n-point (PnP) problem [3,11,14,15,17]. The goal of this problem is to find the absolute pose of the camera and its intrinsic parameters (the focal length, the optical center, the lens distortion, etc.) from a set of 2D-3D point pairs; it is interpretable as camera calibration from a single image. Several methods using the Gröbner basis technique have been proposed; however, computing the floating-point Gröbner basis has a challenging issue in numerical stability.

In this paper, we propose a novel linear method for solving the problem of the one-sided radial distortion homography. Our strategy is based on the DLT method but estimates the *inverse* of the homography matrix together with the lens distortion parameter. Therefore, we refer to the proposed method as invDLT. The proposed invDLT works for both the minimal (4.5 points) and least-squares (more than five points) cases. Moreover, compared to the conventional methods, invDLT is accurate to image noise, robust to outliers, and easy to implement.

2 Preliminaries

2.1 One-Sided Radial Distortion Homography

In this section, we describe the estimation problem of the homography matrix together with the one-sided radial lens distortion. Figure 1 illustrates the geometric relationship of the problem.

Let us consider a situation where a 3D point $(x, y, 0)$ on the $z = 0$ plane is observed as a 2D point (u, v) by a camera with the radial lens distortion k. Using the division model [8], the undistorted 2D position (u', v') in the homogeneous coordinates is given by

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \frac{1}{1 + kr^2} \begin{bmatrix} u \\ v \end{bmatrix} \longmapsto \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \propto \mathbf{m} = \begin{bmatrix} u \\ v \\ 1 + kr^2 \end{bmatrix}, \tag{1}$$

where $r^2 = u^2 + v^2$ and $\propto$ denotes equality up to scale. Note that, unlike in [8], lens distortion affects only 2D points but not 3D points. Then, the 3×3 homography transformation $\mathtt{H}$ between the 2D and 3D point correspondence can be represented as

$$\mathbf{m} \propto \mathtt{H}\mathbf{x}, \tag{2}$$

where $\mathbf{x} = [x, y, 1]^\mathsf{T}$.

The goal of solving this problem is to find the homography matrix $\mathtt{H}$ and the radial lens distortion k from a set of 2D–3D point correspondences. Since $\mathtt{H}$ is of 3×3 but Eq. 2 holds up to scale, this problem has 9 DoFs in total (eight from $\mathtt{H}$ and one from k). A single point pair gives two constraints, therefore, the problem can be solved if more than 4.5 point correspondences are given.

2.2 Fitzgibbon's Method

Fitzgibbon [8] showed a linear method for finding $\mathtt{H}$ with a single common distortion on both images, i.e., the two-sided distortion case. We apply the Fitzgibbon's approach for the one-sided distortion case in this section.

Equation 2 can be reformulated by

$$\mathbf{m} \times \mathtt{H}\mathbf{x} = \mathbf{0}. \tag{3}$$

Given $N \geq 5$ point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{m}_i; i \in (1, \ldots, N)\}$ and writing $\mathtt{H}$ by a 9-vector $\mathbf{h}$, we obtain

$$(\mathtt{D}_1 - k\mathtt{D}_2)\mathbf{h} = \mathbf{0}, \tag{4}$$

where

$$\mathtt{D}_1 = \begin{bmatrix} \mathbf{0}_{1\times3} & -\mathbf{x}_1^\mathsf{T} & v_1\mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_1^\mathsf{T} & \mathbf{0}_{1\times3} & -u_1\mathbf{x}_1^\mathsf{T} \\ & \vdots & \\ \mathbf{0}_{1\times3} & -\mathbf{x}_N^\mathsf{T} & v_1\mathbf{x}_N^\mathsf{T} \\ \mathbf{x}_N^\mathsf{T} & \mathbf{0}_{1\times3} & -u_1\mathbf{x}_N^\mathsf{T} \end{bmatrix}, \quad \mathtt{D}_2 = \begin{bmatrix} \mathbf{0}_{1\times3} & r_1^2\mathbf{x}_1^\mathsf{T} & \mathbf{0}_{1\times3} \\ -r_1^2\mathbf{x}_1^\mathsf{T} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ & \vdots & \\ \mathbf{0}_{1\times3} & r_N^2\mathbf{x}_N^\mathsf{T} & \mathbf{0}_{1\times3} \\ -r_N^2\mathbf{x}_N^\mathsf{T} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{bmatrix}. \tag{5}$$

The third row of Eq. 3 for each point correspondence is left out as it is a linear combination of the first two, hence yielding the matrices $\mathtt{D}_1$ and $\mathtt{D}_2$ of size $2N \times 9$ rather than $3N \times 9$. Thus, we can find the unknown vector $\mathbf{h}$ and the distortion parameter k by solving the following generalized eigenvalue problem:

$$\mathtt{D}_1^\mathsf{T}\mathtt{D}_1\mathbf{h} = k\mathtt{D}_1^\mathsf{T}\mathtt{D}_2\mathbf{h}. \tag{6}$$

Since the last three columns of $\mathtt{D}_2$ are all zeros, the rank of $\mathtt{D}_1^\mathsf{T}\mathtt{D}_2$ is at most six. The distortion parameter k can be obtained as one of the six real-valued eigenvalues, and the unknown vector $\mathbf{h}$, i.e., the homography matrix $\mathtt{H}$, is the corresponding eigenvector.

Although the above procedure seems correct at first glance, actually it has two issues. First, Eq. 6 is not optimal in the least-squares sense. Equation 6 is derived based on an assumption that the right-hand side is zero in Eq. 4. 2D points contain localization errors in general, resulting in $(\mathtt{D}_1 - k\mathtt{D}_2)\mathbf{h} = \epsilon$. Instead, the following optimization problem must be solved:

$$\begin{aligned} \min_{\mathbf{h},k} \quad & \|(\mathtt{D}_1 - k\mathtt{D}_2)\mathbf{h}\|^2 \\ \text{s.t.} \quad & \|\mathbf{h}\|^2 = 1 \end{aligned} \tag{7}$$

However, the standard DLT approach cannot solve Eq. 7.

The next issue is that there are multiple solutions in Eq. 6, even when $N > 5$. One way to uniquely determine the least-squares solution is to discard k with a large absolute value by heuristic thresholding or to select the best k that gives the minimum reprojection error of other sampling points.

3 Inverse DLT Method

3.1 Basic Formulation

The focus of the proposed method differs from Fitzgibbon's method in that it first estimates the parameters of the backward transformation and then returns to the forward transformation. As shown in Eq. 2 and Fig. 1, the homography matrix $\mathtt{H}$ is a forward transformation from the $z = 0$ plane to the *undistorted* image plane, while the radial lens distortion k is an inverse transformation from the *distorted* observations to the *undistorted* image plane. Hence, it makes sense to simultaneously formulate both $\mathtt{H}$ and k based on the backward transformation.

From Eq. 2, we can write the inverse homography transformation as

$$\mathbf{m} \propto \mathtt{H}\mathbf{x} \;\leftrightarrow\; \mathtt{H}^{-1}\mathbf{m} \propto \mathbf{x}. \tag{8}$$

Now let us introduce the lifted coordinates [2]. The mapping between an undistorted point $\mathbf{m} = [u, v, 1 + kr^2]^\mathsf{T}$ and its lifted coordinate $\hat{\mathbf{m}} = [r^2, u, v, 1]^\mathsf{T}$ can be represented by

$$\begin{bmatrix} u \\ v \\ 1 + kr^2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r^2 \\ u \\ v \\ 1 \end{bmatrix} \;\leftrightarrow\; \mathbf{m} = \mathtt{Q}\hat{\mathbf{m}},\ \mathtt{Q} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

Using Eq. 9, we can equivalently rewrite Eq. 8 as

$$\mathtt{H}^{-1}\mathtt{Q}\hat{\mathbf{m}} \propto \mathbf{x}. \tag{10}$$

If we define a 3×4 matrix $\mathtt{G}$ such that $\mathtt{G} = \mathtt{H}^{-1}\mathtt{Q}$, we obtain

$$\mathbf{x} \times (\mathtt{H}^{-1}\mathtt{Q})\hat{\mathbf{m}} = \mathbf{x} \times \mathtt{G}\hat{\mathbf{m}} = \mathbf{0}. \tag{11}$$

Let $\dot{h}_{i,j}$ be the (i, j) elements of $\mathtt{H}^{-1}$, and $\mathbf{h}_j^{-1}$ the j-th column vector of $\mathtt{H}^{-1}$. We can express $\mathtt{G}$ by

$$\begin{aligned} \mathtt{G} &= \begin{bmatrix} k\dot{h}_{1,3} & \dot{h}_{1,1} & \dot{h}_{1,2} & \dot{h}_{1,3} \\ k\dot{h}_{2,3} & \dot{h}_{2,1} & \dot{h}_{2,2} & \dot{h}_{2,3} \\ k\dot{h}_{3,3} & \dot{h}_{3,1} & \dot{h}_{3,2} & \dot{h}_{3,3} \end{bmatrix} \\ &= \begin{bmatrix} k\mathbf{h}_3^{-1} & \mathtt{H}^{-1} \end{bmatrix}. \end{aligned} \tag{12}$$

Given N point correspondences, we can build a linear equation

$$\mathtt{M}\mathbf{g} = \mathbf{0}, \tag{13}$$

where

$$\mathtt{M} = \begin{bmatrix} \mathbf{0}_{1\times4} & -\hat{\mathbf{m}}_1^\mathsf{T} & y_1\hat{\mathbf{m}}_1^\mathsf{T} \\ \hat{\mathbf{m}}_1^\mathsf{T} & \mathbf{0}_{1\times4} & -x_1\hat{\mathbf{m}}_1^\mathsf{T} \\ & \vdots & \\ \mathbf{0}_{1\times4} & -\hat{\mathbf{m}}_N^\mathsf{T} & y_N\hat{\mathbf{m}}_N^\mathsf{T} \\ \hat{\mathbf{m}}_N^\mathsf{T} & \mathbf{0}_{1\times4} & -x_N\hat{\mathbf{m}}_N^\mathsf{T} \end{bmatrix}, \tag{14}$$

$$\mathbf{g} = [k\dot{h}_{1,3},\ \dot{h}_{1,1},\ \dot{h}_{1,2},\ \dot{h}_{1,3},\ k\dot{h}_{2,3},\ \dot{h}_{2,1},\ \dot{h}_{2,2},\ \dot{h}_{2,3},\ k\dot{h}_{3,3},\ \dot{h}_{3,1},\ \dot{h}_{3,2},\ \dot{h}_{3,3}]^\mathsf{T}. \tag{15}$$

The strategy of the proposed method is to compute the vector $\mathbf{g}$ or the matrix $\mathtt{G}$ from the design matrix $\mathtt{M}$, which is of size $2N \times 12$. Since $\mathtt{G}$ has 9 DoFs and a single point pair (Eq. 11) gives two constraints, we can solve $\mathbf{g}$ by utilizing $N \geq 4.5$ points, where 0.5 means one of the two constraints.

3.2 Minimal Solution ($N = 4.5$ Points)

Given five point correspondences, we obtain the 10×12 matrix $\mathtt{M}$ from Eq. 14. Since the minimal case is 4.5 points as mentioned in Sect. 3.1, we select any (non-degenerate) 9 rows out of 10 of $\mathtt{M}$ and say it as $\mathtt{M}_{9\times12}$. The 12-vector $\mathbf{g}$ can be parameterized as a linear combination of the nullspace vectors $\mathbf{n}_i$ of $\mathtt{M}_{9\times12}$:

$$\mathbf{g} \propto \alpha_1\mathbf{n}_1 + \alpha_2\mathbf{n}_2 + \alpha_3\mathbf{n}_3 = \mathtt{N}\boldsymbol{\alpha}, \tag{16}$$

where α_i are unknown coefficients, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3]^\mathsf{T}$, and $\mathtt{N} = [\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3]$. One of α_i is typically set to one, e.g., $\alpha_3 = 1$, to eliminate the scale ambiguity of $\mathbf{g}$ [12]. It contributes to reduce the number of unknowns; however, it also causes a numerical degeneracy if $\alpha_3 \approx 0$ [14]. Instead, we utilize a more sophisticated way to determine α_i.

From Eq. 15, we can see that $[g_1, g_5, g_9]$ and $[g_4, g_8, g_{12}]$ are linearly dependent, where g_i is the i-th element of $\mathbf{g}$. Those six elements can be written in the form

$$\begin{bmatrix} g_1 \\ g_5 \\ g_9 \\ g_4 \\ g_8 \\ g_{12} \end{bmatrix} = \begin{bmatrix} k\dot{h}_{1,3} \\ k\dot{h}_{2,3} \\ k\dot{h}_{3,3} \\ \dot{h}_{1,3} \\ \dot{h}_{2,3} \\ \dot{h}_{3,3} \end{bmatrix} = \begin{bmatrix} n_{1,1} & n_{1,2} & n_{1,3} \\ n_{5,1} & n_{5,2} & n_{5,3} \\ n_{9,1} & n_{9,2} & n_{9,3} \\ n_{4,1} & n_{4,2} & n_{4,3} \\ n_{8,1} & n_{8,2} & n_{8,3} \\ n_{12,1} & n_{12,2} & n_{12,3} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \tag{17}$$

where $n_{i,j}$ denotes the (i, j) element of $\mathtt{N}$. Representing Eq. 17 in a matrix form, we have

$$\begin{bmatrix} k\mathbf{h}_3^{-1} \\ \mathbf{h}_3^{-1} \end{bmatrix} = \begin{bmatrix} \mathtt{A}_{3\times 3} \\ \mathtt{B}_{3\times 3} \end{bmatrix} \boldsymbol{\alpha}, \tag{18}$$

where $\mathtt{A}$ and $\mathtt{B}$ are the first and the last three rows of the nullvector matrix in Eq. 17, respectively. We can rewrite Eq. 18 by

$$\begin{aligned} k\mathbf{h}_3^{-1} &= \mathtt{A}\boldsymbol{\alpha}, \\ \mathbf{h}_3^{-1} &= \mathtt{B}\boldsymbol{\alpha}. \end{aligned} \tag{19}$$

Thus, we can obtain k and $\boldsymbol{\alpha}$ by solving the following generalized eigenvalue problem:

$$\mathtt{A}\boldsymbol{\alpha} = k\mathtt{B}\boldsymbol{\alpha}. \tag{20}$$

Since $\mathtt{A}$ and $\mathtt{B}$ are 3×3 matrices, Eq. 20 can be solved easily and stably. There are at most three real solutions of k and $\boldsymbol{\alpha}$.

Substituting $\boldsymbol{\alpha}$ into Eq. 16, we can obtain $\mathbf{g}$. According to Eq. 12, we can finally recover the *forward* homography matrix $\mathtt{H}$ by taking the inverse of the last three columns of $\mathtt{G}$. That is, letting $\mathbf{g}_j$ be the j-th column of $\mathtt{G}$, we can compute $\mathtt{H}$ by

$$\mathtt{H} = \begin{bmatrix} \mathbf{g}_2 \ \mathbf{g}_3 \ \mathbf{g}_4 \end{bmatrix}^{-1}. \tag{21}$$

3.3 Least Squares Solution ($N > 5$ Points)

In the least squares case, i.e., $N > 5$ point correspondences, Eq. 11 does not hold for all point pairs due to image noise. Consequently, $\mathtt{M}\mathbf{g} = \epsilon$. Since the row rank of $\mathtt{M}$ is 12 for noisy points in general (11 if noise-free), we can find $\mathbf{g}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{g}} \quad & \|\mathtt{M}\mathbf{g}\|^2 \\ \text{s.t.} \quad & \|\mathbf{g}\|^2 = 1 \end{aligned} \tag{22}$$

Equation 22 is a form of the standard DLT method, therefore, we can determine $\mathbf{g}$ as the eigenvector associated with the smallest eigenvalue of $\mathtt{M}^\mathsf{T}\mathtt{M}$.

After obtained the vector $\mathbf{g}$ and its matrix form $\mathtt{G}$, we can recover the homography matrix $\mathtt{H}$ as shown in Eq. 21. Moreover, we can calculate k by

$$k = \frac{\mathbf{g}_1^\top \mathbf{g}_4}{\|\mathbf{g}_4\|^2}. \tag{23}$$

Since the element g_{12}, or the (3,3) element of $\mathtt{H}^{-1}$, is never zero for plausible homography transformations, the denominator is always $\|\mathbf{g}_4\| > 0$. Moreover, the numerator is $\mathbf{g}_1^\top \mathbf{g}_4 = 0$ if $k = 0$. Hence, Eq. 23 holds for any k.

4 Experiment

In this section, we report experimental evaluations of the proposed method compared to the conventional ones. First, we tested the proposed method on synthetic data to investigate the numerical stability, the robustness against image noise, and the performance in the presence of outliers. Then, we evaluated the performance of the methods using feature points obtained from real images.

We have implemented the following methods on MATLAB:

invDLT ($N \geq 4.5$) The proposed minimal and least-squares methods for finding the homography matrix with radial distortion (Sect. 3). At most three solutions for the minimal case and a single solution for the least-square case.

AWF ($N \geq 5$) A minimal and least-squares methods by Fitzgibbon's method [8] for finding the homography matrix with radial distortion (Sect. 2.2). At most six solutions for the minimal and the least-squares cases.

p4pfr ($N = 4$) A Gröbner basis minimal solver proposed by Larsson et al. [14] for solving the absolute camera pose problem with unknown focal length and unknown radial distortion. At most 13 solutions for the minimal case.

VPnPfr ($N \geq 6$) A Gröbner basis least-squares solver proposed by Nakano [17] for solving the perspective-n-point problem with unknown focal length and unknown radial distortion. At most 20 solutions for the least-squares case.

We have conducted all experiments on a PC with Core i7-13700K.

4.1 Synthetic Data Evaluation

We synthesized 3D scenes to conduct quantitative evaluations. We randomly set a single camera of which Euler angles were $-30° \leq \theta_x \leq 30°$, $-30° \leq \theta_y \leq 30°$, $-180° \leq \theta_z \leq 180°$ and translation components were $-1 \leq t_x \leq 1$, $-1 \leq t_y \leq 1$, $2 \leq t_z \leq 6$. The image resolution was 1920×1080, and the optical center (or the center of the distortion) was set to the image center, i.e., $[c_x, c_y] = [960, 540]$. The focal length of the camera was set to $f = 960$, which corresponds to the 90° horizontal field of view. Hence, we calculated the ground-truth homography matrix by $\mathtt{H}_{\mathrm{gt}} = \mathtt{K}[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}]$, where $\mathtt{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{r}_1$ and $\mathbf{r}_2$

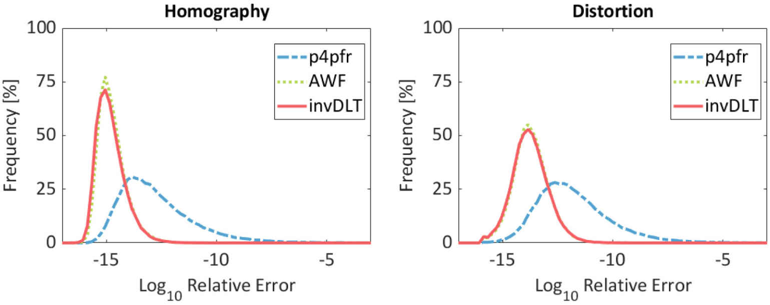


Fig. 2. Numerical accuracy of the minimal solvers for noiseless data. The difference of AWF and invDLT is not visible because the two lines are almost overlapped.

Table 1. Runtime comparison of the minimal solvers on the synthetic data

Method	Time [msec]
p4pfr	0.3664
AWF	0.0683
invDLT	0.0534

are the first and second columns of the rotation matrix, and $\mathbf{t} = [t_x, t_y, t_z]^\top$. We varied the radial distortion k_{gt} in each experiment. We generated N pairs of a 2D–3D point correspondence as follows: 1) randomly chose N distorted 2D points (u, v) within $[1920, 1080]$, 2) computed undistorted coordinates (u', v') by Eq. 1, 3) determined $z = 0$ planar points $(x, y, 0)$ by back-projection, $\mathtt{H}^{-1}[u', v', 1]^\top$.

Numerical Errors and Runtime in Minimal Case. We first measured numerical errors of the three minimal solvers, invDLT, AWF, and p4pfr. We generated 100000 scenes and randomly set the radial distortion $-0.01/f^2 \leq k_{\text{gt}} \leq -0.4/f^2$ for each trial. Figure 2 shows the histograms of Log$_{10}$ relative errors between the ground-truth and estimated values: $|k_{\text{gt}} - k_{\text{est}}|/|k_{\text{gt}}|$ and $\min(\|\mathtt{H}_{\text{gt}} - \mathtt{H}_{\text{est}}\|_{\text{Fro}}, \|\mathtt{H}_{\text{gt}} + \mathtt{H}_{\text{est}}\|_{\text{Fro}})$, where the two matrices were normalized so that their Frobenius norms were one. We can see that AWF and the proposed invDLT have almost the same performance and are more accurate than p4pfr. This is because invDLT and AWF utilize simple linear operations, while p4pfr requires a 40×50 Gaussian elimination and a 13×13 eigenvalue problem.

The complexity of the methodology also affects the computational time. Table 1 shows the average runtime of each solver. The proposed invDLT is the fastest, even 20% faster than AWF. Moreover, both methods are more than five times faster than p4pfr.

Accuracy w.r.t. Varying Image Noise. In this experiment, we investigated the robustness of the three least-squares solvers (invDLT, AWF, VPnPfr) against

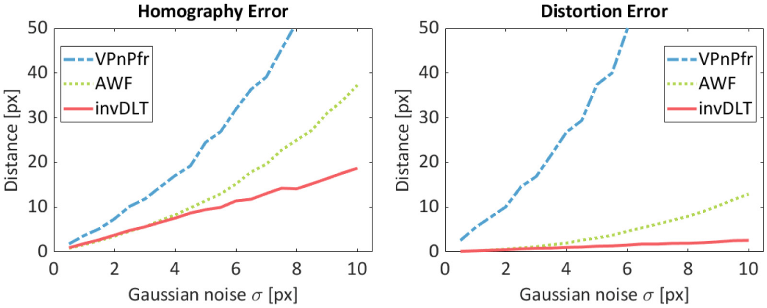


Fig. 3. Accuracy w.r.t. varying image noise.

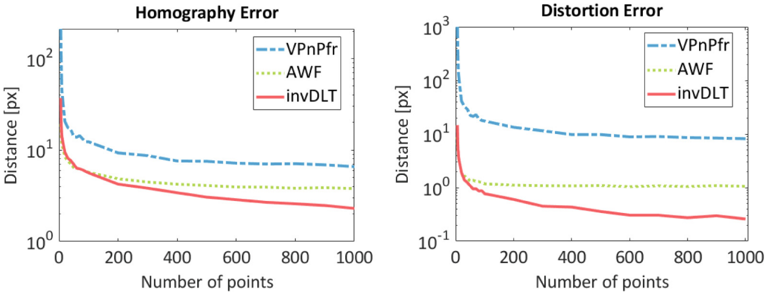


Fig. 4. Accuracy w.r.t. varying number of the points.

image noise. We fixed the radial distortion by $k_{\mathrm{gt}} = -0.2/f^2$. We generated $N = 100$ points and added the zero-mean Gaussian noise on the points by varying the standard deviation ranging $0.5 \leq \sigma \leq 10$. We measured the homography estimation error by calculating the average corner loss [6], i.e., transforming the four corner points surrounding the 3D planar points with $\mathtt{H}_{\mathrm{gt}}$ and $\mathtt{H}_{\mathrm{est}}$, respectively, and then calculating the average L2 distance between the true and estimated 2D corners. Also, we measured the distortion estimation error as the average distance of 2D positions that were undistorted using k_{gt} and k_{est} with Eq. 1. We uniformly sampled 2D points for undistortion by dividing the image coordinates into 50×50 blocks.

Figure 3 shows the median errors over 1000 independent trials for each noise level. The estimation error by invDLT is the most moderate for the noise increase. In particular, the difference against AWF becomes more significant for $\sigma \leq 4$. These results indicate that invDLT is the most robust against image noise.

Accuracy w.r.t. Varying Number of Points. We tested the estimation accuracy of the three least-squares methods for changes in the number of point correspondences. We used the fixed distortion $k_{\mathrm{gt}} = -0.2/f^2$ and the fixed noise level $\sigma = 2$. We varied the number of the points, $6 \leq N \leq 1000$, and measured the average corner loss and distortion errors.

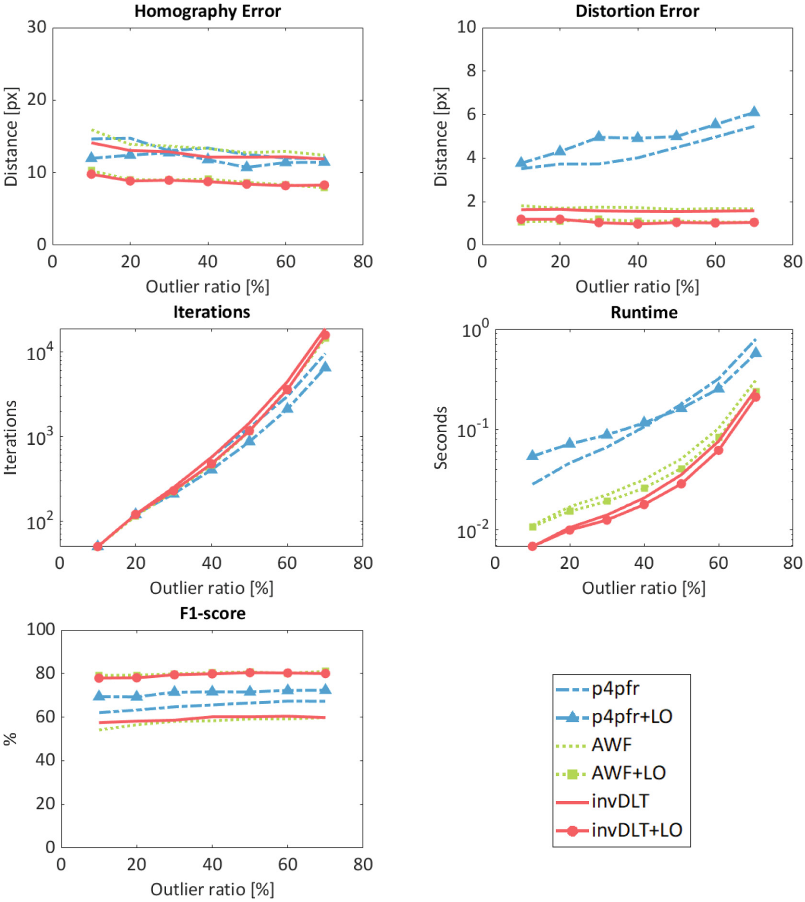


Fig. 5. Synthetic data evaluation in the presence of outliers. Note that the curves of invDLT and AWF are almost overlapped.

Figure 4 shows the median errors over 1000 independent trials for each N. The invDLT's estimation error decreases as N increases whereas that of VPnPfr and AWF reach a plateau.

Accuracy w.r.t. Varying Outlier Ratio. We studied the performance in the presence of the outliers, i.e., the input point pairs are contaminated by wrong matches. We implemented the vanilla RANSAC [7] with the three minimal solvers and incorporated the three least-squares solvers into LO-RANSAC [5]. Since p4pfr is a 4-point minimal solver, we combined VPnPfr into the LO-RANSAC with p4pfr as the least-squares method. We configured $k_{\text{gt}} = -0.2/f^2$, $N = 1000$, $\sigma = 2$, and the inlier-outlier threshold by 3 pixels. We varied the outlier ratio by $0.1 \leq w \leq 0.7$, then set the maximum number of iterations by

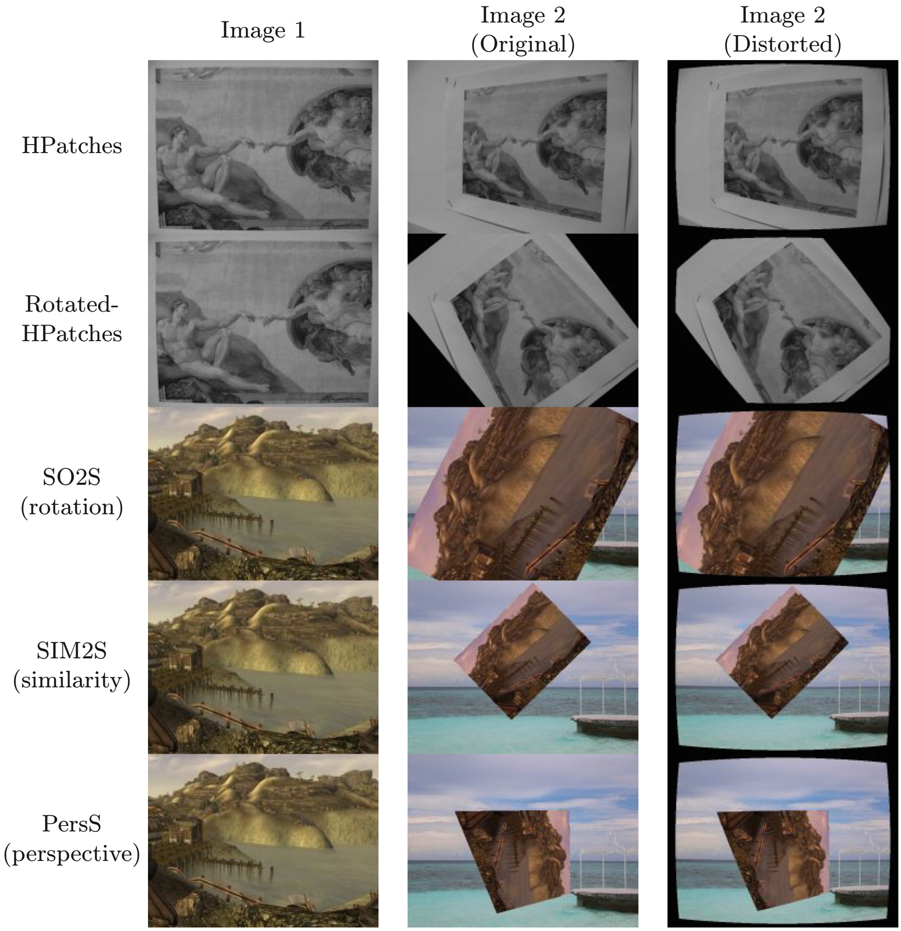


Fig. 6. Image pairs of the datasets.

$10 \times \log(1 - 0.99)/\log(1 - (1 - w)^5)$ to ensure that the RANSACs can find a solution. After convergence, we applied the final refinement using the least-square method with all predicted inliers to polish the accuracy of the solution. We updated the upper limit of the iterations each time more inliers were found. In this experiment, in addition to the homography and distortion errors, we measured the runtime, the number of iterations until convergence, and the F1-score of predicted inliers.

Figure 5 shows the median values of the five metrics over 1000 independent trials for each outlier ratio. The LO-step accurately obtains more inliers and significantly reduces the estimation error by invDLT and AWF. The minimal solver of invDLT is faster than that of AWF as described in Sect. 4.1, and as a result, the overall computational time of invDLT becomes shorter in the RANSAC scenarios. Moreover, it is notable that invDLT utilizing 5-points is faster than p4pfr.

Table 2. Results of AUC on the HPatches, Rotated-HPathces, and SIM2E datasets

Dataset	Method	Homography error			Distortion error		
		@5px	@10px	@20px	@5px	@10px	@20px
HPatches	p4pfr	0.08	0.14	0.22	0.13	0.20	0.29
	AWF	0.24	0.36	0.49	0.35	0.46	0.58
	invDLT	**0.30**	**0.45**	**0.60**	**0.56**	**0.71**	**0.82**
Rotated-HPatches	p4pfr	0.02	0.06	0.18	0.07	0.12	0.19
	AWF	0.18	0.31	0.44	0.29	0.40	0.52
	invDLT	**0.25**	**0.41**	**0.55**	**0.54**	**0.70**	**0.81**
SO2S (rotation)	p4pfr	0.46	0.61	0.71	0.37	0.49	0.61
	AWF	0.34	0.56	0.71	0.44	0.62	0.75
	invDLT	**0.56**	**0.73**	**0.82**	**0.89**	**0.91**	**0.92**
SIM2S (similarity)	p4pfr	0.08	0.18	0.30	0.11	0.21	0.33
	AWF	0.30	0.46	0.58	0.24	0.34	0.45
	invDLT	**0.35**	**0.52**	**0.64**	**0.67**	**0.73**	**0.77**
PersS (perspective)	p4pfr	0.00	0.02	0.06	0.03	0.06	0.10
	AWF	0.06	0.15	0.26	0.09	0.18	0.29
	invDLT	**0.27**	**0.43**	**0.55**	**0.62**	**0.68**	**0.72**

4.2 Real Data Evaluation

Finally, we evaluated the proposed invDLT and the two conventional methods on real image datasets using feature point detection and matching. We used three publicly available datasets: HPatches [1], Rotated-HPatches [18], and SIM2E [20]. HPatches consists of 59 planar scenes, providing 354 image pairs. Rotated-HPatches is a modification of HPatches, which adds a significant 2D rotational change to each image pair. SIM2E is a CG-based dataset composed of three subsets of geometric transformations: SO2S (only 2D rotations), SE2S (similarity transformations), and PersS (perspective transformations). Each subset consists of 71 planar objects, providing 1482 image pairs. All datasets provide the ground-truth homography matrix $\mathtt{H}_{\mathrm{gt}}$ for each image pair. Dataset images are shown in Fig. 6.

We conducted the experiments as follows. For each image pair, we defined the first image as the $z = 0$ plane and the second as an image taken from a different viewpoint. Setting the focal length f as the 80% of the image width and the lens distortion $k_{\mathrm{gt}} = -0.2/f^2$, we simulated a radial distortion on the second images. Then, we obtained SIFT [16] feature matches from the image pairs[1]. Finally, we evaluated the average corner error and the distortion error by applying the LO-RANSAC, of which threshold and the maximum iterations were set by 3 pixels and 5000, respectively, on the initial point matches.

[1] We used a SIFT implementation on VL-Feat: https://www.vlfeat.org/.

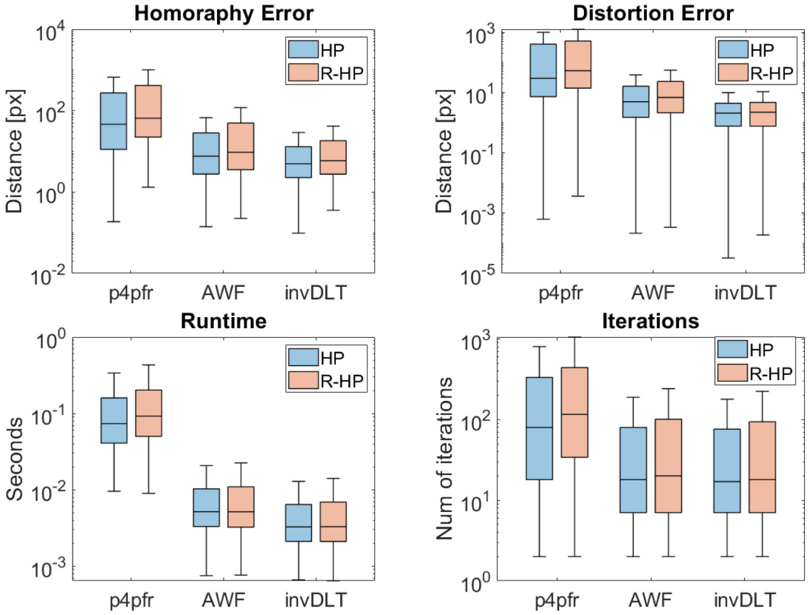


Fig. 7. Quantitative results on the HPatches and Rotated-HPatches datasets.

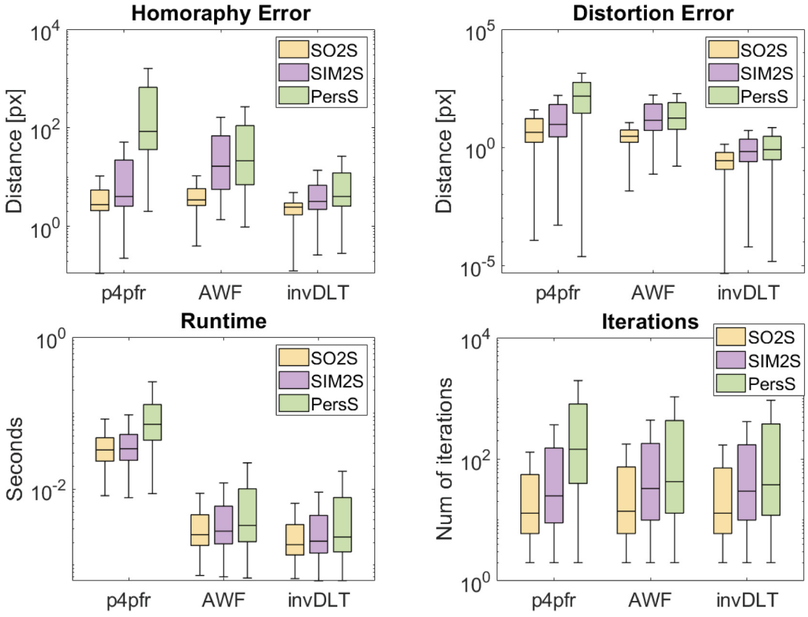


Fig. 8. Quantitative results on the SIM2E dataset.

We used the same random seed between all methods for a single trial and conducted 100 independent trials for each image pair. Table 2 reports the area under the cumulative curve (AUC) at 5, 10, 20 pixels for the two error metrics. The AUC result clearly indicates that the proposed invDLT provides more accurate estimations than p4pfr and AWF in all thresholds. Figures 7 and 8 show the detailed distributions of the experimental results by box plots. In each plot, the box indicates 25% to 75% quartiles, the horizontal bar shows the median, and the whiskers indicate the 1.5 interquartile ranges. The proposed invDLT has more minor statistical variance than p4pfr and AWF in all criteria, and the median values by invDLT are also lower. Notably, despite using 4.5 points, invDLT is much faster by orders of magnitude than p4pfr while achieving more accuracy. From these observations, we can conclude that invDLT is superior to the conventional methods in practice.

5 Conclusion

In this paper, we have introduced a novel method for estimating the planar homography along with one-sided radial lens distortion. By considering the inverse transformation, we derived an optimal formulation for handling noisy data and presented a straightforward DLT-based method applicable to both the minimal and least-squares cases. Extensive experiments with synthetic and real data demonstrated that the proposed invDLT significantly outperforms conventional methods, achieving higher accuracy and robustness while maintaining faster computational times.

References

1. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: a benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5173–5182 (2017)
2. Barreto, J.P., Daniilidis, K.: Fundamental matrix for cameras with radial distortion. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 1, pp. 625–632. IEEE (2005)
3. Bujnak, M., Kukelova, Z., Pajdla, T.: New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In: Asian Conference on Computer Vision, pp. 11–24. Springer (2010)
4. Byröd, M., Brown, M., Åström, K.: Minimal solutions for panoramic stitching with radial distortion. In: The 20th British Machine Vision Conference. British Machine Vision Association (BMVA) (2009)
5. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Pattern Recognition: 25th DAGM Symposium, Magdeburg, Germany, September 10–12, 2003. Proceedings 25, pp. 236–243. Springer (2003)
6. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep Image Homography Estimation (2016). arXiv preprint: arXiv:1606.03798

7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
8. Fitzgibbon, A.: Simultaneous linear estimation of multiple view geometry and lens distortion. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–I (2001). 10.1109/CVPR.2001.990465
9. Guan, B., Vasseur, P., Demonceaux, C., Fraundorfer, F.: Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2320–2327. IEEE (2018)
10. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press (2004)
11. Kukelova, Z., Bujnak, M., Pajdla, T.: Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2816–2823 (2013)
12. Kukelova, Z., Bujnak, M., Pajdla, T.: Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In: 2013 IEEE International Conference on Computer Vision, pp. 2816–2823 (2013). https://doi.org/10.1109/ICCV.2013.350
13. Kukelova, Z., Heller, J., Bujnak, M., Pajdla, T.: Radial distortion homography. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 639–647 (2015)
14. Larsson, V., Kukelova, Z., Zheng, Y.: Making minimal solvers for absolute pose estimation compact and robust. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2316–2324 (2017)
15. Larsson, V., Sattler, T., Kukelova, Z., Pollefeys, M.: Revisiting radial distortion absolute pose. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1062–1071 (2019)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**, 91–110 (2004)
17. Nakano, G.: A versatile approach for solving pnp, pnpf, and pnpfr problems. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14, pp. 338–352. Springer (2016)
18. Parihar, U.S., Gujarathi, A., Mehta, K., Tourani, S., Garg, S., Milford, M., Krishna, K.M.: Rord: Rotation-robust descriptors and orthographic views for local feature matching. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1593–1600. IEEE (2021)
19. Schmalstieg, D., Hollerer, T.: Augmented Reality: Principles and Practice. Addison-Wesley Professional (2016)
20. Su, S., Zhao, Z., Fei, Y., Li, S., Chen, Q., Fan, R.: Sim2e: benchmarking theÂ group equivariant capability ofÂ correspondence matching algorithms. In: Karlinsky, L., Michaeli, T., Nishino, K. (eds.) Computer Vision - ECCV 2022 Workshops, pp. 743–759. Springer Nature Switzerland, Cham (2023)
21. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)

Best of Both Sides: Integration of Absolute and Relative Depth Sensing Modalities Based on iToF and RGB Cameras

I-Sheng Fang[1], Wei-Chen Chiu[2](✉), and Yong-Sheng Chen[2]

[1] Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
[2] National Yang Ming Chiao Tung University, Hsinchu, Taiwan
walon@cs.nctu.edu.tw

Abstract. LiDAR sensors have become one of the most popular active depth sensing devices nowadays with their wide applications in autonomous driving and robotics. Among various types of LiDARs, indirect time of flight (iToF) has been ubiquitously applied on smartphones and consumer-level imagining devices due to its affordable price. Based on the common camera configuration on nowadays smartphones of having an iToF sensor and multiple RGB cameras with different focal lengths (thus leading to different fields of view), in this work, we investigate the integration between two opposite but complementary sensing modalities to achieve better depth estimation: 1) The active sensing modality based on iToF provides absolute and metric depths but suffers from noises caused by environmental lighting and heat; 2) The passive sensing modality based on monocular RGB cameras produces high-resolution but relative depth estimation. Our proposed integration is built upon a weakly-supervised learning framework where the learning objective mainly stems from the inter-camera geometric consistency with the help of iToF depth estimates. Moreover, we adopt the structure distillation technique for preserving structure details from the passive sensing method. We conduct experiments on both synthetic and real-world datasets and demonstrate that the depth estimation produced by the proposed integration model has a comparable quantitative performance with respect to the supervised learning baselines. Besides, the qualitative evaluation of our model shows that it utilizes the advantages and further overcomes the limitations of both sensing modalities.

Keywords: Multiple view geometry · Multi-modal and multi-view learning · Stereo and 3D vision

I-Sheng Fang—Work done at NYCU as graduate student.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78444-6_30.

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 463–479, 2025.
https://doi.org/10.1007/978-3-031-78444-6_30

1 Introduction

Depth estimation is an essential task in computer vision. Among various depth sensors, RGB-D camera modules attract attention because of their capability of multimodal perception from the environment, providing the depth and the RGB images simultaneously. For the RGB-D camera module of consumer-level mobile phones, time-of-flight (ToF) cameras are the more affordable solution. As shown in Fig. 1, the camera module used in this study comprises an indirect time-of-flight (iToF) depth camera, an ultra-wide-angle RGB camera, and a wide-angle RGB camera. Our objective is to obtain accurate metric depth with the same field of view (FoV) as that of the RGB image.

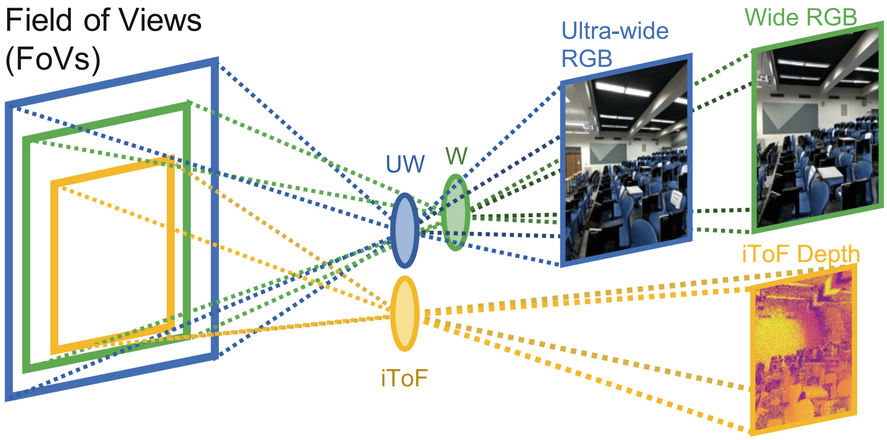


Fig. 1. Illustration of our RGB-D camera module with specific emphasis on the differences in terms of focal lengths and fields of view (FoV). Compared with the RGB cameras, the iToF camera typically has smaller resolution and longer focal length, leading to the narrower FoV.

As shown in Fig. 2, we have the active sensing depths measured by the iToF camera and the passive sensing depths estimated from the RGB image by the off-the-shelf vision-based monocular depth estimation model [19]. The iToF depth camera measures the phase shift between the emitted and reflected infrared light [10] for depth calculation. As a result, the depth measured by iToF is accurate in short range and has metric (absolute) values. However, its resolution and FoV are relatively lower than those of the depth maps estimated from RGB images. As shown in the right column of Fig. 2, the iToF depths warped onto the RGB image plane have a large invalid part with void values (yellow region with depth value 0). Moreover, the iToF depths suffer from different types of noises and errors, such as multi-path interference errors, periodic noises, and low reflection of the infrared signal, causing inaccurate warping results. On the other hand, vision-based monocular depth estimation models [23,31] have shown impressive performance in the depth estimation with high-resolution results [19]. These models benefit from the variety of large datasets and the learned depth cues of objects, such as edges and vanishing points [12]. However, the obtained depths are relative values and may suffer from incorrect depth cues due to the domain gap. In short, the active and passive depth sensing modalities are complementary to each other and their integration stands a good chance in the combination of advantages from both sides. Our goal is to obtain a metric depth map with high resolution and less noise by utilizing both iToF depths and RGB images.

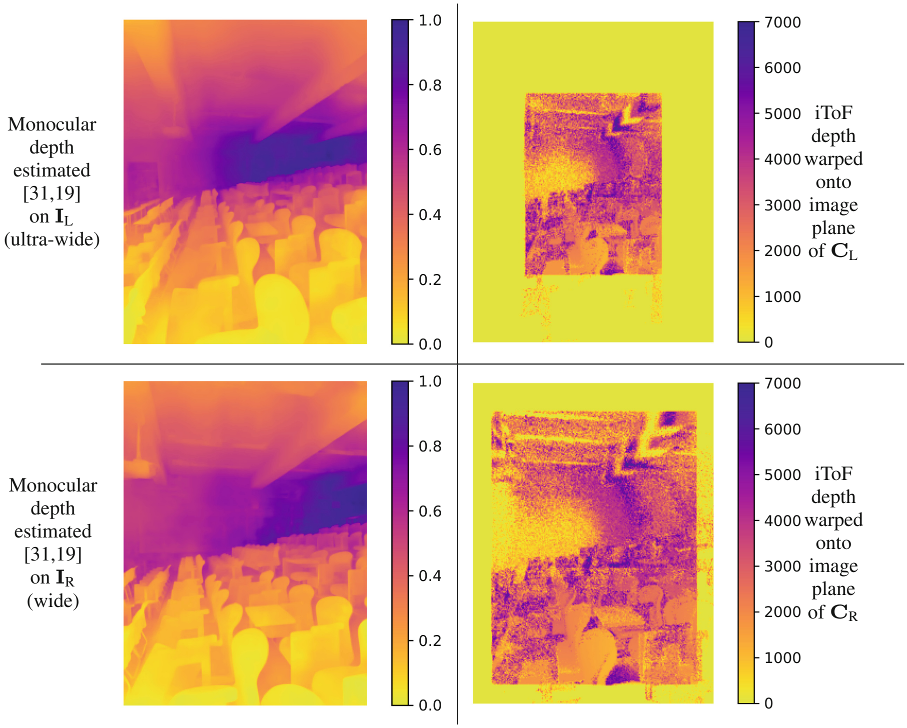
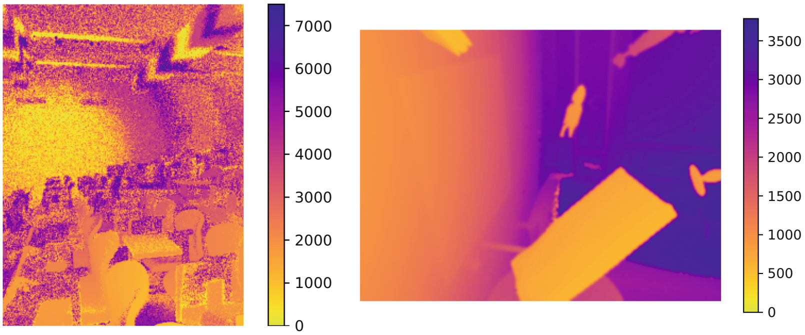


Fig. 2. An example set of the monocular depth estimation [19,31] on $\mathbf{I}_\mathrm{L}$ and $\mathbf{I}_\mathrm{R}$, with the corresponding iToF depth maps $\{\mathbf{D}^\mathrm{L}_\mathrm{iToF}, \mathbf{D}^\mathrm{R}_\mathrm{iToF}\}$ being warped onto the image planes of their respective cameras (*i.e.* $\mathbf{C}_\mathrm{L}$ with ultra-wide-angle lens and $\mathbf{C}_\mathrm{R}$ with wide-angle lens). Notice that $\{\mathbf{D}^\mathrm{L}_\mathrm{iToF}, \mathbf{D}^\mathrm{R}_\mathrm{iToF}\}$ stemmed from iToF sensor have metric depth values with smaller FoVs and contain more noises, whereas the depth maps computed by the off-the-shelf monocular depth estimation model [19,31] have higher resolutions but only relative depth values.

The straightforward idea for cross-modal depth integration is to utilize the confidence map of the metric depth, filter out the unreliable depth measurements, and train the model with supervised learning as a depth completion task. However, our iToF depth camera lacks the information for uncertainty, making it difficult to expose the confident regions in the iToF depth map. Moreover, it is difficult to reduce the influence of noise using RANSAC [30] because of the large amounts of noises in iToF depth map. Therefore, iToF depths cannot be used as ground truths for supervised learning. Furthermore, although the structured light [28] could obtain the ground-truth depths, it is labor-intensive and sensitive to noise. Another way for supervised learning is to adopt synthetic data [21]. Unfortunately, the problem of the domain gap between the real and synthetic images is difficult to overcome. As shown in Fig. 3, iToF depths taken by our device have high-frequency and periodic noises, which are not typical in the synthetic dataset ToF-FlyingThings3D [21], causing the issue of deployment in the real world.



iToF-RGB1k

ToF-FlyingThings3D [21]

Fig. 3. Comparison on the iToF depth maps between our collected iToF-RGB1k dataset and the synthetic ToF-FlyingThings3D dataset [21]. The noises of iToF depth maps in iToF-RGB1k are high-frequency and periodic while those in ToF-FlyingThings3D are low-frequency and non-periodic, leaving a large domain gap to deploy the model trained on synthetic data.

To tackle these challenges, we propose a cross-modal depth estimation model to integrate passive sensing RGB and active sensing iToF images as well as its weakly supervised learning method. Instead of direct supervision with ground-truth depths, the training of our model is self-supervised with the consistency of multi-view geometry by computing the similarity between the captured RGB image and the warped one according to the estimated depths. Moreover, our model leverages the off-the-shelf monocular depth estimation model to extend the original limited FoV of iToF and distill the knowledge of depth structure.

In summary, contributions of this work include:

1. We propose a cross-modal depth estimation model and its weakly-supervised learning framework containing the **cross-warp consistency** and **depth**

structure distillation. This model integrates the active iToF depths with the passive RGB image to obtain the metric depth map having the same FoV as the RGB image.

2. We collect the real-world dataset iToF-RGB1k with 1074 sets of triplet data for the training and testing of the cross-modal depth estimation model. Each triplet contains an ultra-wide RGB image, a wide RGB image, and an iToF depth map.
3. Quantitative evaluation using the synthetic dataset ToF-FlyThings3D[21] as training data shows that our model gains competitive results compared with other supervised learning methods, even though our model is a weakly supervised learning method. Our model also qualitatively performs well when trained and tested on real-world dataset iToF-RGB1k.

2 Related Works

2.1 Depth Completion

The objective of depth completion is to estimate a dense and accurate depth map from a sparse or incomplete one by recovering missing or invalid depth values. Ma et al. [17] propose the Sparse-to-Dense method to predict the dense depth map from a sparse set of depth measurements and a single RGB image. In their following work, Ma et al. [16] further improve Sparse-to-Dense by utilizing photometric consistency and camera poses calculated by PnP with RANSAC. Wong et al. [29] and Choi et al. [3] utilize temporal photometric consistency with pose estimation network and L_1 loss. DFuseNet [26] utilizes stereo photometric consistency in depth completion task. While these approaches fill in missing depth values based on confident measurements, our method extends the FoV of depth images without confidence filtering. Moreover, our model tackles the problem of large FoV differences among three cameras without additional pose estimation or stereo image rectification.

2.2 iToF Depth Refinement and Cross-Modal Depth Estimation

Because of the success of deep learning [14] in various machine learning tasks, many network models have been proposed to refine iToF depth, requiring synthetic data for supervised learning [5,9,18,27]. As another modality, RGB has been used for iToF refinement or depth estimation with supervised learning for model training [13,21]. CroMo method [28] utilizes geometric consistency for self-supervised learning from the cross-modal dataset with iToF and stereo polarization images. Instead of depth data used in our method, CroMo uses iToF correlation images. Moreover, their stereo RGB cameras are with the same focal length, but ours are different. Furthermore, our method distills the knowledge of depth structure from other off-the-shelf monocular depth estimation models.

2.3 Monocular Depth Estimation and Knowledge Distillation

Monocular depth estimation models use the visual depth cue to estimate the spatial relationship between objects [12] from a single image. Godard et al. [7] introduce a self-supervised-learning method with left-right consistency. Recent supervised-learning works, such as MiDaS [23], DPT [22], and LeReS [31], leverage neural networks with advanced model structures and large diverse datasets. Miangoleh et al. [19] discover the trade-off between scene structure and high-frequency details and mix the estimated depths with low and high resolutions to boost the performance of the off-the-shelf model. Inspired by knowledge distillation, DistDepth [30] distills depth-domain structure knowledge from the off-the-shelf model into its monocular depth estimator. In contrast with our method which integrates RGB and iToF modalities to estimate absolute depths, these works use single modality (RGB) and most of them estimate relative depths.

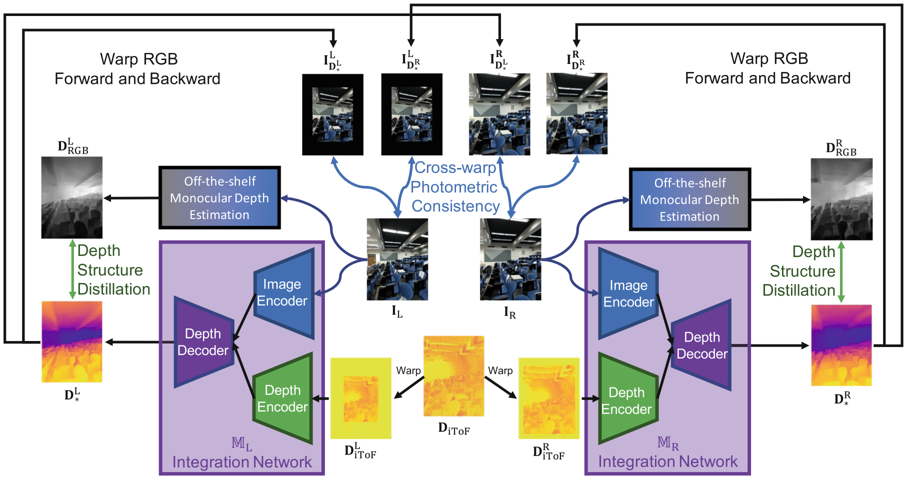


Fig. 4. Computational flow of our framework of integrating active and passive depth sensing modalities (*i.e.* iToF sensor and RGB cameras respectively). We warp the iToF depth map onto the image planes of the RGB cameras for rough alignment. Then, we input the RGB image and the warped iToF depth map into the integration network to integrate modalities and to estimate metric depth on the perspective of the RGB image. Integration Network is weakly supervised by cross-warp consistency and depth structure distillation. See Sect. 3 for details.

3 Methods

3.1 Problem Statement

Our cross-modal integration scenario for depth estimation is built upon a RGB-D camera module composed of:

1. Left RGB camera $\mathbf{C}_{\mathrm{L}}$ with an ultra-wide-angle lens, where the image captured by $\mathbf{C}_{\mathrm{L}}$ is denoted as $\mathbf{I}_{\mathrm{L}}$;
2. Right RGB camera $\mathbf{C}_{\mathrm{R}}$ with a wide-angle lens, where the image captured by $\mathbf{C}_{\mathrm{R}}$ is denoted as $\mathbf{I}_{\mathrm{R}}$;
3. iToF depth camera $\mathbf{C}_{\mathrm{iToF}}$, in which $\mathbf{C}_{\mathrm{iToF}}$ produces the iToF depth map $\mathbf{D}_{\mathrm{iToF}}$.

In this RGB-D camera module, the FoV of $\mathbf{I}_{\mathrm{L}}$ is larger than $\mathbf{I}_{\mathrm{R}}$, and the iToF camera is typically with the smallest FoV. Without loss of generality, we assume that the camera with ultra-wide-angle lens is placed on the left of the one with wide-angle lens. The objective of our cross-modal integration is to acquire the depth maps $\mathbf{D}_{\mathrm{L}}$ and $\mathbf{D}_{\mathrm{R}}$ respectively for both $\mathbf{C}_{\mathrm{L}}$ and $\mathbf{C}_{\mathrm{R}}$, with well taking the complementary properties between $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ and $\mathbf{C}_{\mathrm{iToF}}$ to achieve the better depth perception results. The depth map $\mathbf{D}_{\mathrm{L}}$ is expected to consist of absolute-metric and less-noisy depths from the same perspective of the left RGB camera $\mathbf{C}_{\mathrm{L}}$.

3.2 Camera Calibration

Prior to realizing cross-modal integration of the RGB-D camera module, we calibrate all the cameras to get their geometric characteristics (*i.e.* intrinsic parameters $\mathbf{K}_{\mathrm{L}}$, $\mathbf{K}_{\mathrm{R}}$, and $\mathbf{K}_{\mathrm{iToF}}$ for $\mathbf{C}_{\mathrm{L}}$, $\mathbf{C}_{\mathrm{R}}$, and $\mathbf{C}_{\mathrm{iToF}}$ respectively) as well as their geometric relationship (*i.e.* extrinsic parameters $\mathbf{T}_{\mathrm{iToF}\rightarrow\mathrm{L}}$, $\mathbf{T}_{\mathrm{L}\rightarrow\mathrm{iToF}}$, $\mathbf{T}_{\mathrm{iToF}\rightarrow\mathrm{R}}$, $\mathbf{T}_{\mathrm{R}\rightarrow\mathrm{iTof}}$, $\mathbf{T}_{\mathrm{L}\rightarrow\mathrm{R}}$, and $\mathbf{T}_{\mathrm{R}\rightarrow\mathrm{L}}$ between cameras, where $\mathbf{T}_{\mathrm{iToF}\rightarrow\mathrm{L}}$ denotes the transformation from $\mathbf{C}_{\mathrm{iToF}}$ to $\mathbf{C}_{\mathrm{L}}$ and the others are defined analogously). We adopt the calibration toolkit of OpenCV [2] and a 7×9 metric chessboard pattern to conduct calibration, where the intensity maps of RGB images $\{\mathbf{I}_{\mathrm{L}}, \mathbf{I}_{\mathrm{R}}\}$ and the infrared amplitude map of the iToF camera $\mathbf{C}_{\mathrm{iToF}}$ are taken as inputs.

3.3 Warping iToF Depths and RGB Images

With the extrinsic and intrinsic parameters among RGB and iToF cameras, the **warping grid** for building the pixel-wise correspondence across their image planes now becomes available, under the following computation procedure:

Given the intrinsic parameters $\{\mathbf{K}_{\mathrm{A}}, \mathbf{K}_{\mathrm{B}}\}$ of two cameras $\{\mathbf{C}_{\mathrm{A}}, \mathbf{C}_{\mathrm{B}}\}$, the extrinsic transformation $\mathbf{T}_{\mathrm{A}\rightarrow\mathrm{B}}$ between them, and the depth map D_{A} related to the image plane of $\mathbf{C}_{\mathrm{A}}$, the corresponding pixel p_{B} on the image plane of $\mathbf{C}_{\mathrm{B}}$ for a specific pixel p_{A} on the image plane of $\mathbf{C}_{\mathrm{A}}$ is computed by

$$\mathrm{p}_{\mathrm{B}} = \mathbf{K}_{\mathrm{B}} \mathbf{T}_{\mathrm{A}\rightarrow\mathrm{B}} z_{\mathrm{p}_{\mathrm{A}}} \mathbf{K}_{\mathrm{A}}^{-1} \mathrm{p}_{\mathrm{A}} \tag{1}$$

where $z_{\mathrm{p}_{\mathrm{A}}} = \mathrm{D}_{\mathrm{A}}(\mathrm{p}_{\mathrm{A}})$. Based on computing the corresponding pixels across cameras, the forward warping grid from camera $\mathbf{C}_{\mathrm{A}}$ to camera $\mathbf{C}_{\mathrm{B}}$ with the help of depth map $\mathbf{D}_{\mathrm{A}}$ is denoted as $\langle \texttt{proj}(\mathbf{D}_{\mathrm{A}}, \mathbf{T}_{\mathrm{A}\rightarrow\mathrm{B}}, \mathbf{K}_{\mathrm{A}}, \mathbf{K}_{\mathrm{B}}) \rangle$, indicating how the pixels on camera $\mathbf{C}_{\mathrm{A}}$'s image plane should move in order to be aligned with the content on the image plane of camera $\mathbf{C}_{\mathrm{B}}$ (following the similar notations as

DistDepth [30]). Moreover, we denote the backward warping grid from camera $\mathbf{C}_{\mathrm{B}}$ to camera $\mathbf{C}_{\mathrm{A}}$ (*i.e.* the inverse mapping with respect to the forward warping grid) as $\langle \mathtt{proj}(\mathbf{D}_{\mathrm{A}}, \mathbf{T}_{\mathrm{A}\rightarrow\mathrm{B}}, \mathbf{K}_{\mathrm{A}}, \mathbf{K}_{\mathrm{B}})\rangle^{-1}$.

Based on such technique of warping grid, if we treat the iToF depth map $\mathbf{D}_{\mathrm{iToF}}$ itself as a grayscale image on the image plane of iToF camera $\mathbf{C}_{\mathrm{iToF}}$, we then are able to warp it onto the image planes of $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ thus obtaining $\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}$ and $\mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}$ respectively:

$$\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}} = \mathbf{D}_{\mathrm{iToF}} \langle \mathtt{proj}(\mathbf{D}_{\mathrm{iToF}}, \mathbf{T}_{\mathrm{iToF}\rightarrow\mathrm{L}}, \mathbf{K}_{\mathrm{iToF}}, \mathbf{K}_{\mathrm{L}})\rangle \tag{2}$$

$$\mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}} = \mathbf{D}_{\mathrm{iToF}} \langle \mathtt{proj}(\mathbf{D}_{\mathrm{iToF}}, \mathbf{T}_{\mathrm{iToF}\rightarrow\mathrm{R}}, \mathbf{K}_{\mathrm{iToF}}, \mathbf{K}_{\mathrm{R}})\rangle \tag{3}$$

in which $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}\}$ seem to already provide the depth perception from the perspective of $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$. However, as iToF cameras typically have a longer focal length than the RGB ones thus leading to the narrower FoV, the warped depth maps (*i.e.* $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}\}$) from iToF camera $\mathbf{C}_{\mathrm{iToF}}$ to RGB ones $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ would unfortunately have large void regions. Moreover, the noise on iToF depth map caused by environmental lighting and heat would also lead to the incorrect warping results. Figure 2 shows the void region due to the difference in terms of focal length as well as the wrong warped results caused by iToF noise. Despite these limitations, the benefits of iToF depth, such as active sensing and metric/absolute value, should be preserved after the following integration of RGB and iToF cameras.

3.4 Off-the-Shelf Monocular Depth Estimation

In addition to the warped iToF depth maps $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}\}$, another plausible and popular way of acquiring depth upon the image planes of RGB cameras $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ is to use the off-the-shelf monocular depth estimation model f, thanks to the recent development of (deep-)learning-based techniques. The depth maps $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{RGB}} = f(\mathbf{I}_{\mathrm{L}}), \mathbf{D}^{\mathrm{R}}_{\mathrm{RGB}} = f(\mathbf{I}_{\mathrm{R}})\}$ contribute the largest FoV with respect to $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ (as all the pixels of $\{\mathbf{I}_{\mathrm{L}}, \mathbf{I}_{\mathrm{R}}\}$ have their depth estimates produced by f, while $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}\}$ have quite some void regions) but only produce relative depth perception.

3.5 Integration of RGB and iToF

Given both the active and passive depth sensing components (*i.e.* $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{iToF}}\}$ and $\{\mathbf{D}^{\mathrm{L}}_{\mathrm{RGB}}, \mathbf{D}^{\mathrm{R}}_{\mathrm{RGB}}\}$ respectively) upon the image planes of $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$, we now proceed to integrate them to produce better depth perception. Instead of directly taking $\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}$ and $\mathbf{D}^{\mathrm{L}}_{\mathrm{RGB}}$ as input to the fusion model for producing the final depth estimation where their difference in terms of the depth-scale change would lead to problematic learning, we propose a novel integration framework based on the following learning scheme composed of three important aspects and shown in Fig. 4. Please note that here we take $\mathbf{C}_{\mathrm{L}}$ as an example while $\mathbf{C}_{\mathrm{R}}$ follows the analogous process. 1) An integration network $\mathbb{M}$ (as indicated by the region shaded by light purple color in Fig. 4) adopts the passive sensing RGB image $\mathbf{I}_{\mathrm{L}}$ for refining the active sensing depth component $\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}$ to obtain the refined depth $\mathbf{D}^{\mathrm{L}}_{*}$. The basic idea behind it is leveraging the rich appearance and structure information of the RGB image to help denoising $\mathbf{D}^{\mathrm{L}}_{\mathrm{iToF}}$ as well as enlarging its FoV; 2) To address the lack of ground-truth depth for supervised learning the integration, we leverage the geometric relationship across two RGB cameras $\{\mathbf{C}_{\mathrm{L}}, \mathbf{C}_{\mathrm{R}}\}$ and build the photometric and depth consistency loss to realize the unsupervised learning of $\mathbf{D}^{\mathrm{L}}_{*}$; 3) We adopt the passive component $\mathbf{D}^{\mathrm{L}}_{\mathrm{RGB}}$ as structural guidance for $\mathbf{D}^{\mathrm{L}}_{*}$ during the training of the integration network $\mathbb{M}$. In other words, we distill the knowledge of depth structure from $\mathbf{D}^{\mathrm{L}}_{\mathrm{RGB}}$. These three important aspects in our framework are driven by two main objectives: **cross-warp consistency** and **depth structure distillation**, which we detailed sequentially in the following.

Cross-Warp Consistency. As we tend to maximize the practical usage and the flexibility of our proposed framework, we do not require the training of $\mathbf{D}^{\mathrm{L}}_{*}$ to rely on the ground-truth labels. In other words, the learning of $\mathbf{D}^{\mathrm{L}}_{*}$ is not supervised. Instead, we are inspired by the unsupervised objective built upon the geometric relations between cameras and photometric reconstruction, as proposed by Godard et al. [7], where the accurate depth estimate of the left camera should enable the reconstruction of the right image by warping the left image via the geometric transformation between them. Following the similar idea, we introduce the ***cross-warp photometric consistency loss*** $L^{\mathbf{D}^{\mathrm{L}}_{*}}_{\text{xwarp-I}}$ for the refined depth $\mathbf{D}^{\mathrm{L}}_{*}$:

$$
\begin{aligned}
L^{\mathbf{D}^{\mathrm{L}}_{*}}_{\text{xwarp-I}} &= L^{\mathbf{D}^{\mathrm{L}}_{*}\text{-fwd}}_{\text{xwarp-I}} + L^{\mathbf{D}^{\mathrm{L}}_{*}\text{-bwd}}_{\text{xwarp-I}} \\
&= \mathbb{S}(\mathbf{I}^{\mathrm{R}}_{\mathbf{D}^{\mathrm{L}}_{*}}, \mathbf{I}_{\mathrm{R}}) + \mathbb{S}(\mathbf{I}^{\mathrm{L}}_{\mathbf{D}^{\mathrm{L}}_{*}}, \mathbf{I}_{\mathrm{L}})\,, \\
\text{where } \mathbb{S}(a,b) &= \alpha \frac{1 - \mathtt{SSIM}(a,b)}{2} + (1-\alpha)\,|a-b|_1 \\
\text{and} \quad \mathbf{I}^{\mathrm{R}}_{\mathbf{D}^{\mathrm{L}}_{*}} &= \mathbf{I}_{\mathrm{L}} \left\langle \mathtt{proj}(\mathbf{D}^{\mathrm{L}}_{*}, \mathbf{T}_{\mathrm{L}\rightarrow\mathrm{R}}, \mathbf{K}_{\mathrm{L}}, \mathbf{K}_{\mathrm{R}}) \right\rangle \\
\mathbf{I}^{\mathrm{L}}_{\mathbf{D}^{\mathrm{L}}_{*}} &= \mathbf{I}_{\mathrm{R}} \left\langle \mathtt{proj}(\mathbf{D}^{\mathrm{L}}_{*}, \mathbf{T}_{\mathrm{L}\rightarrow\mathrm{R}}, \mathbf{K}_{\mathrm{L}}, \mathbf{K}_{\mathrm{R}}) \right\rangle^{-1} .
\end{aligned}
\tag{4}
$$

in which function $\mathbb{S}(a,b)$ evaluates the $\mathtt{SSIM}$ structural distance as well as L_1 pixel errors between a and b (noting that we follow the common practice as [30] to set $\alpha = 0.85$). $\mathbf{I}^{\mathrm{R}}_{\mathbf{D}^{\mathrm{L}}_{*}}$ denotes the reconstructed right image, using $\mathbf{D}^{\mathrm{L}}_{*}$ to perform the forward warping from $\mathbf{C}_{\mathrm{L}}$ to $\mathbf{C}_{\mathrm{R}}$; $\mathbf{I}^{\mathrm{L}}_{\mathbf{D}^{\mathrm{L}}_{*}}$ denotes the reconstructed left image, using $\mathbf{D}^{\mathrm{L}}_{*}$ to perform the backward warping from $\mathbf{C}_{\mathrm{R}}$ to $\mathbf{C}_{\mathrm{L}}$. Noting that $L^{\mathbf{D}^{\mathrm{R}}_{*}}_{\text{xwarp-I}}$ follows the similar procedure to evaluate $\mathbb{S}(\mathbf{I}^{\mathrm{L}}_{\mathbf{D}^{\mathrm{R}}_{*}}, \mathbf{I}_{\mathrm{L}}) + \mathbb{S}(\mathbf{I}^{\mathrm{R}}_{\mathbf{D}^{\mathrm{R}}_{*}}, \mathbf{I}_{\mathrm{R}})$.

In addition to the cross-warp photometric consistency loss $\{L^{\mathrm{L}}_{\text{xwarp-I}}, L^{\mathrm{R}}_{\text{xwarp-I}}\}$ for $\{\mathbf{D}^{\mathrm{L}}_{*}, \mathbf{D}^{\mathrm{R}}_{*}\}$, we also modify the well-known left-right depth consistency loss [7,8] into ***cross-warp depth consistency loss*** $L_{\text{xwarp-D}}$ for our training of integration network $\mathbb{M}$, making the warped depth map of right camera equal to the depth map of left camera and vice versa, regardless of forward or backward warping:

$$
\begin{aligned}
L_{\text{xwarp-D}} &= L^{\mathbf{D}^{\text{L}}_*}_{\text{xwarp-D}} + L^{\mathbf{D}^{\text{R}}_*}_{\text{xwarp-D}} \\
&= L^{\mathbf{D}^{\text{L}}_*\text{-fwd}}_{\text{xwarp-D}} + L^{\mathbf{D}^{\text{L}}_*\text{-bwd}}_{\text{xwarp-D}} + L^{\mathbf{D}^{\text{R}}_*\text{-fwd}}_{\text{xwarp-D}} + L^{\mathbf{D}^{\text{R}}_*\text{-bwd}}_{\text{xwarp-D}} \\
&= \left|\mathbf{D}^{\text{R}}_{\mathbf{D}^{\text{L}}_*} - \mathbf{D}^{\text{R}}_*\right|_1 + \left|\mathbf{D}^{\text{L}}_{\mathbf{D}^{\text{L}}_*} - \mathbf{D}^{\text{L}}_*\right|_1 \\
&+ \left|\mathbf{D}^{\text{L}}_{\mathbf{D}^{\text{R}}_*} - \mathbf{D}^{\text{L}}_*\right|_1 + \left|\mathbf{D}^{\text{R}}_{\mathbf{D}^{\text{R}}_*} - \mathbf{D}^{\text{R}}_*\right|_1 , \\
\text{where } \mathbf{D}^{\text{R}}_{\mathbf{D}^{\text{L}}_*} &= \mathbf{D}^{\text{L}}_* \left\langle \mathtt{proj}(\mathbf{D}^{\text{L}}_*, \mathbf{T}_{\text{L}\rightarrow\text{R}}, \mathbf{K}_{\text{L}}, \mathbf{K}_{\text{R}})\right\rangle , \\
\mathbf{D}^{\text{L}}_{\mathbf{D}^{\text{L}}_*} &= \mathbf{D}^{\text{R}}_* \left\langle \mathtt{proj}(\mathbf{D}^{\text{L}}_*, \mathbf{T}_{\text{L}\rightarrow\text{R}}, \mathbf{K}_{\text{L}}, \mathbf{K}_{\text{R}})\right\rangle^{-1} , \\
\mathbf{D}^{\text{L}}_{\mathbf{D}^{\text{R}}_*} &= \mathbf{D}^{\text{R}}_* \left\langle \mathtt{proj}(\mathbf{D}^{\text{R}}_*, \mathbf{T}_{\text{R}\rightarrow\text{L}}, \mathbf{K}_{\text{R}}, \mathbf{K}_{\text{L}})\right\rangle , \\
\mathbf{D}^{\text{R}}_{\mathbf{D}^{\text{R}}_*} &= \mathbf{D}^{\text{L}}_* \left\langle \mathtt{proj}(\mathbf{D}^{\text{R}}_*, \mathbf{T}_{\text{R}\rightarrow\text{L}}, \mathbf{K}_{\text{R}}, \mathbf{K}_{\text{L}})\right\rangle^{-1} . \qquad (5)
\end{aligned}
$$

Depth Structure Distillation. As motivated previously that our third aspect is to adopt the passive component (e.g. $\mathbf{D}^{\text{L}}_{\text{RGB}}$) as a structural guidance for the output of our integration model, we choose to adapt the ***structure distillation loss*** proposed by [30] into our framework for realizing such aspect, which is defined as

$$
\begin{aligned}
L_{\text{distill}} = &\ L^{\mathbf{D}^{\text{L}}_*}_{\text{distill}} + L^{\mathbf{D}^{\text{R}}_*}_{\text{distill}} \\
= &\ 1 - \mathtt{SSIM}(\bar{\mathbf{D}}^{\mathbf{L}}_*, \bar{\mathbf{D}}^{\mathbf{L}}_{\text{RGB}}) \\
&+ 1 - \mathtt{SSIM}(\bar{\mathbf{D}}^{\mathbf{R}}_*, \bar{\mathbf{D}}^{\mathbf{R}}_{\text{RGB}}) , \qquad (6)
\end{aligned}
$$

where $\bar{\mathbf{D}}$ denotes the operation of normalizing depth $\mathbf{D}$ with respect to its own mean value. The depth structure distillation loss L_{distill} relies on the off-the-shelf pre-trained monocular depth estimation model f to provide the passive depth perception. Therefore, although our cross-warp consistency objective is self-supervised, we categorize our method as a weakly-supervised learning framework.

Smoothness Loss [7]**:** Lastly, similar to other self-supervised depth estimation methods [7,8], we also adopt the ***smoothness loss*** L_{sm} to regulate the estimated depth $\{\mathbf{D}^{\text{L}}_*, \mathbf{D}^{\text{R}}_*\}$ for making them locally smooth and edge-aware:

$$
L_{\text{sm}} = \left|\partial \mathbf{D}^{\text{L}}_*\right| e^{-\|\partial \mathbf{I}_{\text{L}}\|} + \left|\partial \mathbf{D}^{\text{R}}_*\right| e^{-\|\partial \mathbf{I}_{\text{R}}\|}. \qquad (7)
$$

The derivative operation ∂ in L_{sm} includes both the horizontal and vertical gradients.

Total Loss. The overall objective is summarized as:

$$
\begin{aligned}
L_{\text{total}} = &\ \lambda_{\text{xwarp-I}} L^{\mathbf{D}^{\text{L}}_*}_{\text{xwarp-I}} + \lambda_{\text{xwarp-I}} L^{\mathbf{D}^{\text{R}}_*}_{\text{xwarp-I}} \\
&+ \lambda_{\text{xwarp-D}} L_{\text{xwarp-D}} + \lambda_{\text{distill}} L_{\text{distill}} + \lambda_{\text{sm}} L_{\text{sm}} , \qquad (8)
\end{aligned}
$$

where λ hyper-parameters are the weights to balance among the aforementioned losses.

3.6 Integration Network $\mathbb{M}$

Our integration network $\mathbb{M}$ is based on an U-Net [25] architecture which is also similar to the one in monodepth [7]. It contains an image feature encoder, an iToF depth feature encoder, and a feature fusion decoder. For both RGB image feature and iToF depth feature encoders, they adopt ResNet18 [11] as their backbone while the former takes the pretrained weight from ImageNet [4] classification task as warm start. The multi-scale features extracted by layers of both encoders are concatenated in a layer-wise manner and further fed to the corresponding convolutional blocks (of the same scale) in the fusion decoder.

4 Experiments

4.1 Datasets

The experiments are conducted on two datasets: the synthetic *ToF-FlyingThings3D* [21] dataset and the real-world *iToF-RGB1k* dataset collected by ourselves.

ToF-FlyingThings3D [21]. As such dataset is synthetic to have full access to the groundtruth depth, we mainly adopt it for our quantitative evaluation. Two different camera configurations are used in our experiments to synthesize the dataset: 1) **pseudo camera parameters** as used in its original paper [21] for ensuring a fair comparison with other methods, where all the cameras are with the same focal length (thus nearly the same FoV) and the extrinsic transformation is simplified (*i.e.* no rotation and only 2D orthogonal translation); 2) **device camera parameters**, where we adopt the calibration parameters obtained from the RGB-D camera module (*i.e.* the device that we use for collecting our iToF-RGB1k dataset, which has different focal lengths for all three cameras and the extrinsic transformations are more complicated), making the synthesized dataset more challenging for the integration between iToF and RGB cameras.

iToF-RGB1k. We collect such iToF-RGB1k dataset by using the mobile-phone device of RGB-D camera in the natural world, in which it comprises 1074 scenes that have been randomly split into 960 sets for training and 114 sets for testing. The iToF depth has resolution of 640×480, while the RGB images have a resolution of 1280×960. As iToF is better suited for indoor environments, the majority of scenes in the dataset are indoor ones. We also consider the social impact of privacy to avoid capturing the human being.

4.2 Quantitative Experiments

Comparison with iToF Refinement Methods. To ensure a fair comparison with supervised learning methods, we first train our integration network $\mathbb{M}$ using the supervised learning objective proposed by Qiu et al. and follow the same evaluation protocol [21]. This objective is also used in SHARP-Net [5].

Table 1. Comparison with competitive methods on the ToF-FlyingThings3D dataset [21]. SL: Supervised learning.

Methods	SL	Training Ground Truth			Input	Refined	
		Depth		RGB	RGB	Depth	MAE(cm)
		Metric	Relative			FoV	
DeepToF [18]	✓	✓				ToF	4.69
ToF-Net [27]	✓	✓				ToF	4.90
TOF-KPN w/o RGB [21]	✓	✓				ToF	2.44
SHARP-Net [5]	✓	✓				ToF	1.19
TOF-KPN [21]	✓	✓			✓	ToF	1.51
Our network w/TOF-KPN loss [21]	✓	✓			✓	ToF	1.50
Cross-warp				✓	✓	RGB	3.16
Cross-warp + Structure Distillation			✓	✓	✓	RGB	3.01

As shown in the row "our network w/TOF-KPN loss" in Table 1, we successfully reproduce the performance of [21]. We then evaluate the performance of our full model, as shown in the last row in Table 1. Our full model outperforms DeepToF [18] and ToF-Net [27] (both supervised ones) without requiring the strong supervision of ground truth depths and can achieve full FoV of RGB image. Although SHARP-Net [5] has the best performance of mean absolute error (MAE), it is limited to refining the FoV of iToF. Considering the inherited performance gap between the supervised and self-supervised learning methods [7,8], our method performs well as a weakly supervised method.

Ablation Studies on Objectives and Modalities. To investigate how the objectives and input modalities affect the performance of our model, we conduct ablation studies with two camera configurations. As shown in Table 2, our ablation study of objectives starts with the supervised learning baselines (in (a) and (b) rows) and self-supervised learning baselines (in (c) and (d) rows). Then, we use a single RGB image (as shown in (c) rows) or stereo RGB images (as shown in (d) rows) as input for the integration network trained with cross-warp consistency. In pseudo camera configuration, the model using stereo RGB input outperforms the one using monocular RGB input. In device camera configuration, however, the opposite results may be associated with the challenge of warping images and depths with different FoVs. Next, we evaluate the performance of our model with cross-modal stereo input (iToF and RGB), as shown in the (e) rows. The results indicate that the model using cross-modal stereo input outperforms those models using single RGB modality because the iToF depths provide metric information for absolute depth estimation. Lastly, as shown in the (f) rows, model training with structure distillation from passive depths improves most performance metrics (excluding the threshold-based ones), indicating the advantages of better structure guidance and knowledge from the off-the-shelf monocular depth estimation model.

Table 2. Quantitative studies for different supervision and the input. SL: Supervised learning. Weak-sup.: Weakly supervised learning. CW: Cross-warp. SD: Structure distillation. Cam.: Camera. M: Monocular RGB. S: Stereo RGB. KPN: Using supervised depth refinement loss of TOF-KPN [21]. S2D: Using Sparse-to-Dense [17] for supervised learning. L: Left rgb camera. R: right rgb camera. UW: Ultra-wide RGB camera. W: Wide RGB camera. Eval. region: Evaluated region. Ext.: extended FoV of iToF.

#	Input			Weak-sup.			Eval.	Evaluation metrics							
	RGB	iToF	SL	CW	SD	Cam.	Region	MAE(cm) ↓	AbsRel ↓	SqRel ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
(1) Pseudo Camera Parameters															
(a)		✓	KPN			L R	RGB	2.130 1.550	0.04425 10.56	0.3086 215.5	4.475 4.392	0.04165 0.04170	0.9394 0.9747	0.9802 0.9937	0.9957 0.9969
(b)	M	✓	KPN			L R	RGB	1.686 1.212	0.03585 0.03614	0.2254 1.0120	3.541 3.275	0.03512 0.02790	0.9550 0.9770	0.9793 0.9938	0.9958 0.9985
(c)	M			✓		L R	RGB	7.267 5.965	0.08433 0.08375	3.427 2.835	16.30 13.46	0.12330 0.11350	0.9063 0.9105	0.9397 0.9446	0.9543 0.9581
(d)	S			✓		L R	RGB	4.059 5.151	0.05112 0.07691	1.444 2.828	11.04 13.37	0.08002 0.09934	0.9492 0.9250	0.9672 0.9508	0.9753 0.9640
(e)	M	✓		✓		L R	RGB	3.156 3.236	0.05717 0.05914	0.760 1.215	7.775 7.941	0.06248 0.06750	0.9519 0.9448	0.9674 0.9650	0.9762 0.9762
(f)	M	✓		✓	✓	L R	RGB	3.006 3.213	0.04710 0.05669	0.7509 1.255	7.731 7.930	0.06208 0.06841	0.9506 0.9454	0.9651 0.9639	0.9739 0.9764
(g)						L R	iToF	3.005 3.206	0.04502 0.05477	0.7399 1.215	7.786 7.935	0.06098 0.06717	0.9536 0.9475	0.9676 0.9657	0.9764 0.9784
(h)						L R	Ext.	3.536 3.891	0.07709 0.09745	1.259 2.536	7.561 8.328	0.07777 0.08513	0.9140 0.9121	0.9344 0.9350	0.9445 0.9497
(i)	M	✓	S2D			L R	RGB	54.17 56.76	1.383 1.565	118.8 141.3	64.32 66.62	0.4071 0.4281	0.1132 0.1013	0.2456 0.2216	0.4110 0.3746
(j)	M	✓	S2D		✓	L R	RGB	20.52 17.77	0.2246 0.2234	7.323 7.141	27.92 22.99	0.1687 0.1738	0.5968 0.6270	0.8467 0.8390	0.9168 0.9082
(2) Device Camera Parameters															
(a)		✓	KPN			UW W	RGB	5.454 2.473	0.06630 0.03923	1.639 0.5655	13.03 6.923	0.07756 0.04683	0.9333 0.9586	0.9664 0.9861	0.9784 0.9941
(b)	M	✓	KPN			UW W	RGB	1.921 1.604	0.03816 0.02757	0.2513 0.1765	3.655 3.516	0.03778 0.02531	0.9644 0.9774	0.9800 0.9927	0.9893 0.9991
(c)	M			✓		UW W	RGB	6.740 10.60	0.08767 0.14480	2.508 3.976	14.45 17.39	0.10030 0.11650	0.9021 0.8185	0.9846 0.9257	0.9650 0.9576
(d)	S			✓		UW W	RGB	11.62 21.78	0.1270 0.3129	4.356 13.97	22.25 31.36	0.12260 0.17940	0.8578 0.5835	0.9222 0.8098	0.9513 0.8982
(e)	M	✓		✓		UW W	RGB	6.115 9.211	0.07910 0.13140	2.116 2.924	13.42 15.20	0.08497 0.09361	0.9175 0.8456	0.9578 0.9469	0.9747 0.9782
(f)	M	✓		✓	✓	UW W	RGB	5.616 9.376	0.07305 0.12280	1.773 2.995	12.62 15.57	0.07644 0.09220	0.9232 0.8579	0.9607 0.9494	0.9764 0.9752
(g)						UW W	iToF	7.453 8.944	0.09368 0.11670	2.189 2.615	13.42 14.60	0.08332 0.08985	0.9027 0.8718	0.9564 0.9525	0.9764 0.9759
(h)						UW W	Ext.	4.715 10.62	0.06289 0.13980	1.570 4.064	12.03 17.84	0.07208 0.09721	0.9333 0.8194	0.9628 0.9395	0.9764 0.9731

Comparison Between Original and Extended FoVs. As shown in the (g) and (h) rows of Table 2, we evaluate the performance of our estimated depths within the original iToF FoV and the extended region outside the FoV of iToF, as illustrated in Fig. 2. The results with pseudo camera parameters indicate that the estimated depths in the original iToF FoV on both RGB image planes are better than those in the extended FoVs. With device camera parameters, however, this case holds only on the wide-angle image plane, suggesting that the model for the wide-angle cameras could more depend on the metric information from iToF depths than the model for the ultra-wide camera because of the larger overlapping region between iToF and RGB cameras.

Comparison with Depth Completion Method. To align with the setting of Sparse-to-Dense [17], we randomly sample 750 points from $\mathrm{D}_{\mathrm{iToF}}^{\mathrm{L}}$ to generate the sparse depth maps. These sparse depth maps are paired with I_{L} as the training pairs for Sparse-to-Dense [17]. The results in the (i) rows of Table 2, where the worse performance indicates that Sparse-to-Dense [17] is less effective for tackling the noise in iToF depth and for leveraging the complementary properties across modalities. Even being further regularized by the structure distillation loss (as shown in the (j) rows), the performance is still much worse than ours. In contrast, our proposed method leveraging geometric constraints for cross-warp consistency is more effective in alleviating the interference from noisy iToF and gets better fusion results.

4.3 Qualitative Experiments

We conduct experiments using our iToF-RGB1k to qualitatively evaluate our model in the real world. As shown in Fig. 5, our model is capable of extending the original FoV of iToF depths to the FoV of RGB images. Moreover, our model is able to remedy the errors or noises of the iToF sensor. For example, as shown in the first and second row of Fig. 5, the iToF depth values within the circled regions are largely deviated due to the reflection of the wall or the transparent umbrella. Our model refines the results by leveraging the rich appearance and structure information from the RGB image. Other examples shown in the third and fourth row in Fig. 5 demonstrate our model's capability to correct depth errors from the off-the-shelf monocular depth estimation model [19,31]. Monocular depth estimation models, reliant on passive sensing RGB cameras, often misinterpret visual cues [12] from TV screens and walls because of misleading or absent textures. In this case, the depth information from active sensing iToF proves beneficial in resolving the ambiguity. To sum up, our cross-warp and depth structure distillation model successfully integrates the passive sensing RGB image and the active sensing iToF depth to estimate the full FoV metric depth map of the scene.

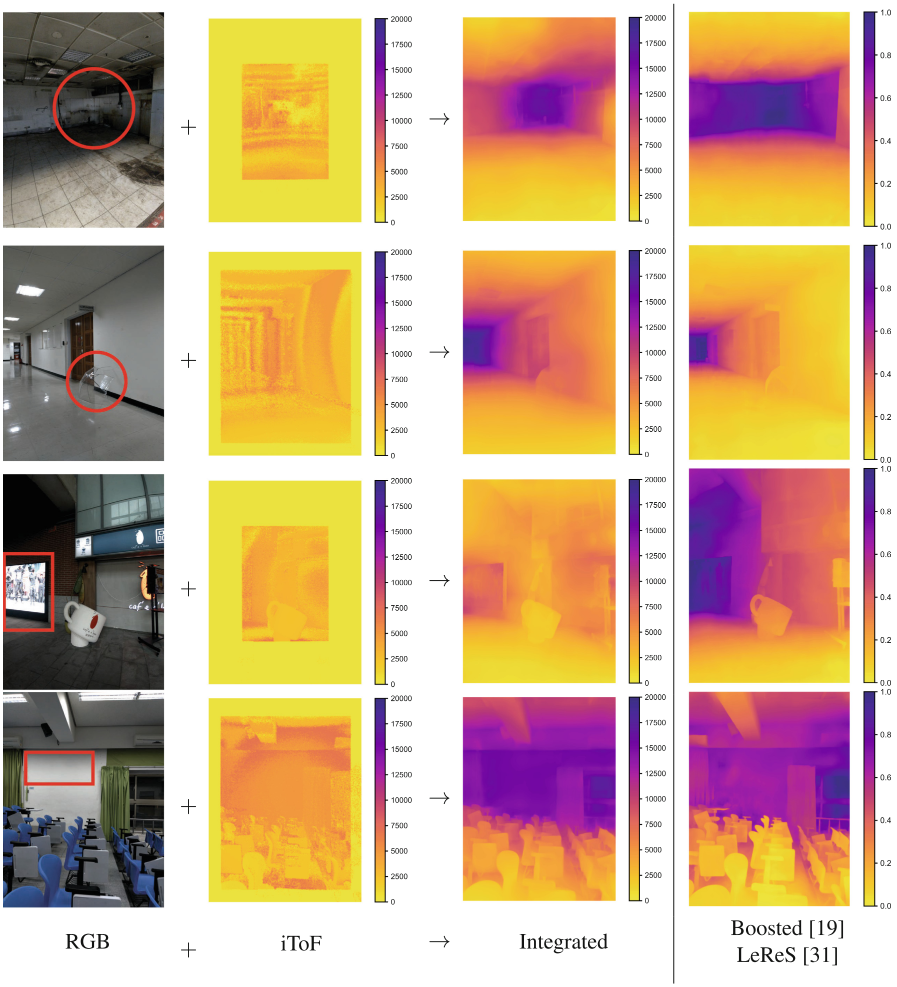


Fig. 5. Qualitative evaluation of estimated depths for the real-world dataset, iToF-RGB1k. Our model overcomes the limitations of the iToF camera and the off-the-shelf monocular depth estimation model, such as the reflective objects and transparent objects (which are red-circled in the first and second row), and the wrong visual depth cues (which are framed by red rectangles in the third and fourth row). Boosted [19] LeReS [31] is the off-the-shelf monocular depth estimation model, a relative depth model. The unit of absolute depth is millimeters. (Color figure online)

5 Conclusions

We introduce a weakly-supervised framework to tackle the task of cross-modal depth estimation, driven by cross-warp consistency and depth structure distillation. Our proposed cross-warp consistency adopts iToF depth estimates to build the inter-camera photometric consistency for guiding the model training, and the depth structure distillation preserves the structure of RGB images under the help of an off-the-shelf monocular depth estimation model. Our quantitative experiment on ToF-FlyThings3D [21] shows that our method is able to achieve comparable performance with several supervised learning methods despite the lack of depth domain ground truths. Moreover, we collect an iToF-RGB1k dataset for performing qualitative evaluation in the real world, in which the corresponding experimental results verify the efficacy of our method in extending the FoV of iToF as well as fixing the incorrect/noisy depth estimate where neither iToF camera nor off-the-shelf monocular depth estimation model can perform well.

References

1. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-Shot Transfer by Combining Relative and Metric Depth (2023). arXiv preprint: arXiv:2302.12288
2. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
3. Choi, J., Jung, D., Lee, Y., Kim, D., Manocha, D., Lee, D.: Selfdeco: self-supervised monocular depth completion in challenging indoor environments. In: ICRA (2021)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR (2009)
5. Dong, G., Zhang, Y., Xiong, Z.: Spatial hierarchy aware residual pyramid network for time-of-flight depth denoising. In: ECCV (2020)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
7. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: CVPR (2017)
8. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: ICCV (2019)
9. Guo, Q., Frosio, I., Gallo, O., Zickler, T., Kautz, J.: Tackling 3D ToF artifacts through learning and the flat dataset. In: ECCV (2018)
10. Hansard, M., Lee, S., Choi, O., Horaud, R.P.: Time-of-Flight Cameras: Principles, Methods and Applications. Springer Science & Business Media (2012)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
12. Hu, J., Zhang, Y., Okatani, T.: Visualization of convolutional neural networks for monocular depth estimation. In: ICCV (2019)
13. Jung, H., Brasch, N., Leonardis, A., Navab, N., Busam, B.: Wild ToFu: improving range and quality of indirect time-of-flight depth with rgb fusion in challenging environments. In: 3DV (2021)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
15. Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Han, J.: On the Variance of the Adaptive Learning Rate and Beyond (2019). arXiv preprint arXiv:1908.03265
16. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: self-supervised depth completion from lidar and monocular camera. In: ICRA (2019)
17. Ma, F., Karaman, S.: Sparse-to-dense: depth prediction from sparse depth samples and a single image. In: ICRA (2018)
18. Marco, J., et al.: Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. ACM TOG (2017)

19. Miangoleh, S.M.H., Dille, S., Mai, L., Paris, S., Aksoy, Y.: Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In: CVPR (2021)
20. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: NeurIPS (2019)
21. Qiu, D., Pang, J., Sun, W., Yang, C.: Deep end-to-end alignment and refinement for time-of-flight RGB-D modules. In: ICCV (2019)
22. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV (2021)
23. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. In: IEEE PAMI (2020)
24. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch. In: WACV (2020)
25. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: MICCAI (2015)
26. Shivakumar, S.S., Nguyen, T., Miller, I.D., Chen, S.W., Kumar, V., Taylor, C.J.: Dfusenet: deep fusion of rgb and sparse depth information for image guided dense depth completion. In: ITSC (2019)
27. Su, S., Heide, F., Wetzstein, G., Heidrich, W.: Deep end-to-end time-of-flight imaging. In: CVPR (2018)
28. Verdié, Y., Song, J., Mas, B., Busam, B., Leonardis, A., McDonagh, S.: Cromo: Cross-modal learning for monocular depth estimation. In: CVPR (2022)
29. Wong, A., Soatto, S.: Unsupervised depth completion with calibrated backprojection layers. In: ICCV (2021)
30. Wu, C.Y., Wang, J., Hall, M., Neumann, U., Su, S.: Toward practical monocular indoor depth estimation. In: CVPR (2022)
31. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3D scene shape from a single image. In: CVPR (2021)

Author Index

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15316, pp. 481–482, 2025.
https://doi.org/10.1007/978-3-031-78444-6