

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

LNCS 15309

# Pattern Recognition

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

9 Part IX

ICPR  
2024 INDIA



 Springer

MOREMEDIA 

# Lecture Notes in Computer Science

15309


## Founding Editors


Gerhard Goos  
Juris Hartmanis

## Editorial Board Members

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

Wen Gao, *Peking University, Beijing, China*

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

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



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


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


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


# Pattern Recognition

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

*Editors*


Apostolos Antonacopoulos   
University of Salford  
Salford, Lancashire, UK

Rama Chellappa   
Johns Hopkins University  
Baltimore, MD, USA

Saumik Bhattacharya   
IIT Kharagpur  
Kharagpur, West Bengal, India

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

Cheng-Lin Liu   
Chinese Academy of Sciences  
Beijing, China

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

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-3-031-78188-9

ISBN 978-3-031-78189-6 (eBook)

<https://doi.org/10.1007/978-3-031-78189-6>

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

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

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

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

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

If disposing of this product, please recycle the paper.

## President's Address

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

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

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

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

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

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

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

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

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

September 2024

Arjan Kuijper  
President of the IAPR

# Preface

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

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

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

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

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

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

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

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

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

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

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

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

October 2024

Umapada Pal  
Josef Kittler  
Anil Jain

# Organization

## General Chairs

Umapada Pal  
Josef Kittler  
Anil Jain

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

## Program Chairs

Apostolos Antonacopoulos  
Subhasis Chaudhuri  
Rama Chellappa  
Cheng-Lin Liu

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

## Publication Chairs

Ananda S. Chowdhury  
Wataru Ohyama

Jadavpur University, India  
Tokyo Denki University, Japan

## Competition Chairs

Richard Zanibbi  
Lianwen Jin  
Laurence Likforman-Sulem

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

## Workshop Chairs

P. Shivakumara  
Stephanie Schuckers  
Jean-Marc Ogier  
Prabir Bhattacharya

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



## **Tutorial Chairs**

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

## **Doctoral Consortium Chairs**

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

## **Organizing Chairs**

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

## **Organizing Committee**

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

## **Finance Chairs**

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

## **Awards Committee Chair**

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

## Sponsorship Chairs

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

## Exhibition and Demonstration Chairs

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

## International Liaison, Visa Chair

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

## Publicity Chairs

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

## Women in ICPR Chairs

Ingela Nystrom	Uppsala University, Sweden
Alexandra B. Albu	University of Victoria, Canada
Jing Dong	Institute of Automation, Chinese Academy of Sciences, China
Sarbani Palit	Indian 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 Lladós	Universitat Autònoma de Barcelona, Spain
Mita Nasipuri	Jadavpur University, India

## Track Chairs – Biomedical Imaging and Bioinformatics

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

## Metareviewers (Conference Papers and Competition Papers)

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

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

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

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

## **Reviewers (Competition Papers)**

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

## Reviewers (Conference Papers)

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

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



Arnav Bhavsar	Bin-Bin Jia
Arnav Varma	Binbin Yong
Arpita Dutta	Bindita Chaudhuri
Arshad Jamal	Bindu Madhavi Tummala
Artur Jordao	Binh M. Le
Arunkumar Chinnaswamy	Bi-Ru Dai
Aryan Jadon	Bo Huang
Aryaz Baradarani	Bo Jiang
Ashima Anand	Bob Zhang
Ashis Dhara	Bowen Liu
Ashish Phophalia	Bowen Zhang
Ashok K. Bhateja	Boyang Zhang
Ashutosh Vaish	Boyu Diao
Ashwani Kumar	Boyun Li
Asifuzzaman Lasker	Brian M. Sadler
Atefeh Khoshkhahtinat	Bruce A. Maxwell
Athira Nambiar	Bryan Bo Cao
Attilio Fiandrotti	Buddhika L. Semage
Avandra S. Hemachandra	Bushra Jalil
Avik Hati	Byeong-Seok Shin
Avinash Sharma	Byung-Gyu Kim
B. H. Shekar	Caihua Liu
B. Uma Shankar	Cairong Zhao
Bala Krishna Thunakala	Camille Kurtz
Balaji Tk	Carlos A. Caetano
Balázs Pálffy	Carlos D. Martá-Nez-Hinarejos
Banafsheh Adami	Ce Wang
Bang-Dang Pham	Cevahir Cigla
Baochang Zhang	Chakravarthy Bhagvati
Baodi Liu	Chandrakanth Vipparla
Bashirul Azam Biswas	Changchun Zhang
Beiduo Chen	Changde Du
Benedikt Kottler	Changkun Ye
Beomseok Oh	Changxu Cheng
Berkay Aydin	Chao Fan
Berlin S. Shaheema	Chao Guo
Bertrand Kerautret	Chao Qu
Bettina Finzel	Chao Wen
Bhavana Singh	Chayan Halder
Bibhas C. Dhara	Che-Jui Chang
Bilge Günsel	Chen Feng
Bin Chen	Chenan Wang
Bin Li	Cheng Yu
Bin Liu	Chenghao Qian
Bin Yao	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  
Dipanjoyoti Paul  
Divya Biligere Shivanna  
Divya Saxena  
Divya Sharma  
Dmitrii Matveichev  
Dmitry Minskiy  
Dmitry V. Sorokin  
Dong Zhang  
Donghua Wang  
Donglin Zhang  
Dongming Wu  
Dongqiangzi Ye  
Dongqing Zou  
Dongrui Liu  
Dongyang Zhang  
Dongzhan Zhou  
Douglas Rodrigues  
Duarte Folgado  
Duc Minh Vo  
Duoxuan Pei  
Durai Arun Pannir Selvam  
Durga Bhavani S.  
Eckart Michaelsen  
Elena Goyanes  
Élodie Puybareau

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

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

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

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

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

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

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



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

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

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

Sneha Shukla  
 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 Napol on  
 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 B chner

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

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

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

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

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



## Contents – Part IX

Mask and Compress: Efficient Skeleton-Based Action Recognition in Continual Learning .....	1
<i>Matteo Mosconi, Andriy Sorokin, Aniello Panariello, Angelo Porrello, Jacopo Bonato, Marco Cotogni, Luigi Sabetta, Simone Calderara, and Rita Cucchiara</i>	
Text-Driven Prototype Learning for Few-Shot Class-Incremental Learning .....	16
<i>Seongbeom Park, Haeji Jung, Daewon Chae, Hyunju Yun, Sungyoon Kim, Suhong Moon, Jinkyu Kim, and Seunghyun Park</i>	
Dual Supervised Contrastive Learning Based on Perturbation Uncertainty for Online Class Incremental Learning .....	32
<i>Shibin Su, Zhaojie Chen, Guoqiang Liang, Shizhou Zhang, and Yanning Zhang</i>	
Breaking Information Silos: Global Guided Task Prediction for Class-Incremental Learning .....	48
<i>Chaoshun Hu, Biaohua Ye, Zixuan Chen, and Jian-Huang Lai</i>	
Conditioned Prompt-Optimization for Continual Deepfake Detection .....	64
<i>Francesco Laiti, Benedetta Liberatori, Thomas De Min, and Elisa Ricci</i>	
Plasticity Driven Knowledge Transfer for Continual Deep Reinforcement Learning in Financial Trading .....	80
<i>Dimitrios Katsikas, Nikolaos Passalis, and Anastasios Tefas</i>	
Orthogonal Latent Compression for Streaming Anomaly Detection in Industrial Vision .....	94
<i>Han Gao, Huiyuan Luo, Fei Shen, and Zhengtao Zhang</i>	
Out-of-Distribution Forgetting: Vulnerability of Continual Learning to Intra-class Distribution Shift .....	111
<i>Liangxuan Guo, Yang Chen, and Shan Yu</i>	
Generating Multi-objective Fronts from Streamed Data Using Nested List .....	128
<i>Arnabi Mukherjee, Sourab Mandal, and Paramartha Dutta</i>	
Mapping the Unknown: A New Approach to Open-World Video Recognition .....	144
<i>César D. Parga, Xosé M. Pardo, and Carlos V. Regueiro</i>	

ESL: Explain to Improve Streaming Learning for Transformers .....	160
<i>Meghna P. Ayyar, Jenny Benois-Pineau, and Akka Zemhari</i>	
Detection of Unknown Errors in Human-Centered Systems .....	176
<i>Aranyak Maity, Ayan Banerjee, and Sandeep K. S. Gupta</i>	
Source-Free Test-Time Adaptation For Online Surface-Defect Detection .....	192
<i>Yiran Song, Qianyu Zhou, and Lizhuang Ma</i>	
Alleviating Catastrophic Forgetting in Facial Expression Recognition with Emotion-Centered Models .....	208
<i>Israel A. Laurensi, Alceu de Souza Britto Jr., Jean Paul Barddal, and Alessandro Lameiras Koerich</i>	
Satellite State Prediction and Maneuver Detection Analysis Using NCDEs .....	225
<i>Kangjun Lee and Simon S. Woo</i>	
MIXAD: Memory-Induced Explainable Time Series Anomaly Detection .....	242
<i>Minha Kim, Kishor Kumar Bhaumik, Amin Ahsan Ali, and Simon S. Woo</i>	
Rough Set Theoretic Approach for Solving the Multi-Armed Bandit Problems .....	258
<i>Avinash Paidi, Istapriya Jagravi, and Pabitra Mitra</i>	
Hybrid Graph Representation Learning: Integrating Euclidean and Hyperbolic Space .....	276
<i>Lening Li, Lei Luo, and Yanguang Sun</i>	
Learning Object Focused Attention .....	291
<i>Vivek Trivedy, Amani Almalki, and Longin Jan Latecki</i>	
Stereographic Projection for Embedding Hierarchical Structures in Hyperbolic Space .....	307
<i>Shangyu Chen, Xiaohao Yang, Pengfei Fang, Mehrtash Tafazzoli Harandi, Dinh Phung, and Jianfei Cai</i>	
SPCSE: Soft Positive Enhanced Contrastive Learning for Sentence Embeddings .....	322
<i>Lingen Liu, Zixin Chen, and Guang Chen</i>	
Neural Topic Model with Distance Awareness .....	337
<i>Shangyu Chen, He Zhao, Viet Huynh, Dinh Phung, and Jianfei Cai</i>	
Ontology-Guided Deep Metric Learning and Applications to Obstetrics .....	353
<i>Jules Bonnard, Arnaud Dapogny, Ferdinand Dhombres, and Kévin Bailly</i>	

CoFE: Consistency-Driven Feature Elimination for eXplainable AI ..... 368  
*Revoti Prasad Bora, Philipp Terhörst, Raymond Veldhuis,  
Raghavendra Ramachandra, and Kiran Raja*

From One to Many Lorikeets: Discovering Image Analogies in the CLIP  
Space ..... 383  
*Songlong Xing, Elia Peruzzo, Enver Sangineto, and Nicu Sebe*

A Framework for Mining Collectively-Behaving Bots in MMORPGs ..... 400  
*Hyunsoo Kim, Jun Hee Kim, Jaeman Son, Jihoon Song, and Eunjo Lee*

Causal Deep Learning ..... 420  
*M. Alex O. Vasilescu*










Non-symmetrical Confidence Interval of AUC Measure Based  
on Cross-Validation ..... 439  
*Yu Wang, Xiaoyan Zhao, and Xingli Yang*

Visualizing and Generalizing Integrated Attributions ..... 455  
*Ethan Payne, David Patrick, and Amanda S. Fernandez*

**Author Index** ..... 471



# Mask and Compress: Efficient Skeleton-Based Action Recognition in Continual Learning

Matteo Mosconi<sup>1</sup>(✉) , Andriy Sorokin<sup>1</sup> , Aniello Panariello<sup>1</sup> ,  
Angelo Porrello<sup>1</sup> , Jacopo Bonato<sup>2</sup> , Marco Cotogni<sup>2</sup> , Luigi Sabetta<sup>2</sup> ,  
Simone Calderara<sup>1</sup> , and Rita Cucchiara<sup>1</sup> 

<sup>1</sup> AImageLab - University of Modena and Reggio Emilia, Modena, Italy

[matteo.mosconi@unimore.it](mailto:matteo.mosconi@unimore.it)

<sup>2</sup> Leonardo S.p.A., Rome, Italy

**Abstract.** The use of skeletal data allows deep learning models to perform action recognition efficiently and effectively. Herein, we believe that exploring this problem within the context of Continual Learning is crucial. While numerous studies focus on skeleton-based action recognition from a traditional offline perspective, only a handful venture into online approaches. In this respect, we introduce CHARON (Continual Human Action Recognition On skeletoNs), which maintains consistent performance while operating within an efficient framework. Through techniques like uniform sampling, interpolation, and a memory-efficient training stage based on masking, we achieve improved recognition accuracy while minimizing computational overhead. Our experiments on NTU-60 and the proposed NTU-120 datasets demonstrate that CHARON sets a new benchmark in this domain. The code is available at <https://github.com/Sperimetal3/CHARON>.

**Keywords:** Continual Learning · Skeleton Based Action Recognition · Class Incremental Learning · Masked Autoencoder

## 1 Introduction

**Human Action Recognition (HAR)** has become critical in various domains such as surveillance [27, 29], rehabilitative healthcare [51], and sports analysis [23, 39]. Early HAR approaches focused on exploiting RGB or gray-scale videos due to their widespread availability. However, recent advancements have explored alternative modalities, including skeletal joints [10, 25, 51], depth [36], point clouds [15], acceleration [24], and WiFi signals [42]. Among these, **skeleton-based action recognition** stands out as particularly efficient and concise, especially for actions not involving objects or scene context. Skeleton sequences capture the trajectory of key points (*i.e.*, joints) in the human body (*e.g.*, elbows, knees, wrists) [48]. As joints can be represented by 2D or 3D spatial

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_1](https://doi.org/10.1007/978-3-031-78189-6_1).

coordinates, skeletal data offer greater efficiency than images due to the sparsity of skeleton graphs. Moreover, this data structure is robust against changes in appearance, cluttered backgrounds, and occlusion while inherently privacy-preserving [42].

The traditional learning approach to HAR assumes that all necessary data is readily available during training. However, this assumption often does not hold in real-world contexts, as instances or classes may emerge incrementally over time. In such a dynamic context, Deep Neural Networks struggle to acquire new knowledge, often displacing the capabilities acquired during the previous stages. This phenomenon – widely known as *catastrophic forgetting* – leads to worse performance and is the focal point of **Continual Learning (CL)**. Specifically, in the CL setting, models must adapt to address a series of tasks presented sequentially, preserving performance on previously seen ones.

While tasks such as classification [5, 22, 35, 41, 46] and video-based action recognition [6, 30, 45] have been widely explored in a Continual Learning setting, skeleton-based HAR has been the subject of limited study in this domain. Although the authors of [26] have made efforts to address this task, they employ an expandable architecture, which can append a new learnable module to the network each time a new class arises. While such a technique aids in alleviating catastrophic forgetting, the computational footprint of the model gradually grows, making the approach memory-hungry and poorly scalable. Additionally, their setting adds constraints that diverge from real-world scenarios. Namely, they pre-train the network on most training instances and retain only a few classes for the incremental stage.

In this work, we exploit the structure of skeletal data to efficiently store samples in an episodic memory, *i.e.*, a continuously updated *buffer* containing a small subset of past data. Specifically, we enhance the memory efficiency of each sample, thus expanding the effective capacity of the buffer within the same memory allocation. We can do so as skeleton sequences present redundancy in time [23], so they can be compressed by sampling a subset of skeletal poses (*e.g.*, only one every  $s$  frames). This operation reduces the temporal resolution of the sequence with minimal information loss. Finally, in later tasks, we reconstruct each retained sample through linear interpolation, which remarkably does not require additional parameters.

We further exploit the redundancy of skeleton sequences by leveraging an approach based on Masked Image Modeling (MIM) [2, 17, 43]. Such self-supervised pre-training techniques have recently gained popularity due to the reduced wall-clock time and memory footprint. These methods pre-train a network by feeding it only a portion of the input data and reconstructing it with a lightweight decoder module. Once the pre-training is completed, they discard the decoder and feed the entire input to the model. However, unlike previous works [47, 49], which employ masking techniques on skeletal data only for pre-training, our approach jointly optimizes both the self-reconstruction and the recognition tasks. Such a choice brings two benefits: *i)* the training time and memory requirements remarkably decrease, and *ii)* the additional reconstruction task acts as a regularizer for the encoder, leading to more meaningful representations.

Finally, at the end of each task, we introduce a *linear probing* phase to better conciliate the self-reconstruction approach with online scenarios. Indeed, if no countermeasures are involved, the encoder may suffer from a covariate shift issue [19] during inference, as it has been trained only on a portion of the input but is tested on the whole data. As reported in Sect. 3.2, this may be heavily detrimental to the final classification layer, specifically for high masking ratios. To mitigate such a problem, we freeze the encoder parameters and re-align the classifier in the presence of unmasked input sequences. This process is remarkably lightweight (*i.e.*, optimizing less than 4K parameters for NTU-60), yet significantly enhances overall performance.

To assess the proposed approach, we conduct a comprehensive evaluation on the incremental version of two popular datasets, NTU RGB+D 60 [38] and NTU RGB+D 120 [28], achieving state-of-the-art performance for class-incremental action recognition in the skeletons domain.

We remark on the following main contributions:

- We reduce the memory requirements of skeleton sequences in the buffer.
- We introduce a MIM approach for efficiently handling skeletal data in CL.
- We employ a linear probing phase to seamlessly integrate the encoder-decoder approach to the incremental learning setting.

## 2 Related Works

**Skeleton-Based Action Recognition.** In early skeleton-based action recognition works, sequences were treated as time series, thus processed employing Recurrent Neural Networks (RNNs) [8, 11, 18, 53] to capture dynamics over time. These approaches struggled to integrate the spatial context of joints and proved slow and challenging to parallelize. Following works exploited Convolutional Neural Networks (CNNs) [20, 21], treating skeletal data in various ways to make them compatible with CNNs; some handle coordinates as image channels [10, 25], while others reshape skeletons by combining joints in space and time [20].

However, these models faced a common limitation: they failed to effectively represent the relationships between skeletal joints moving together in time. Graph Convolutional Networks (GCNs) resolve such shortcomings by exploiting nodes (*i.e.*, joints) temporally and spatially [7, 12, 13, 40, 50]. Subsequently, the emergence of ViT [9] marked the introduction of transformer-based architectures into computer vision, leading to solutions that integrate self-attention layers into convolutional architectures. One such work, STTFormer [31], divides the sequence in tuples of joints and retains some concepts of CNNs (*i.e.*, pooling aggregation) for in-time features processing. Nonetheless, such an approach under-exploits the sparsity and redundancy of skeletal data. In recent years, masking approaches [47, 49] have been employed to take advantage of these characteristics for pre-training models. In contrast, our proposal adopts the reconstruction objective even during the optimization of the downstream task. Such a choice brings the benefit of reducing the training requirements of the whole pipeline, avoiding the pre-training phase.

**Continual Learning.** The Continual Learning setting makes a more realistic assumption w.r.t. standard learning paradigms. Specifically, data arrival is continuous and incremental. A subset of CL is Class-Incremental Learning (Class-IL) [44], where the dataset is re-arranged into multiple subsequent tasks, each containing a unique and disjoint set of classes. In this setting, the task identity is not known during inference.

Classical CL methods employ a regularization term that penalizes the alterations of weights to avoid forgetting [22, 37, 52]. Rehearsal methods [1, 5, 32, 34], on the other hand, employ a limited memory buffer in which they store samples from past tasks and replay them. Another paradigm is represented by dynamic architectures [3, 35] in which new network components are instantiated for each incoming task; unfortunately, this often leads to a rapid increase in the number of parameters. This approach has been employed by the authors of Else-Net [26] to tackle skeleton-based HAR in Class-IL. They use the first 50 classes of NTU RGB+D 60 to pre-train their network, and perform incremental training across 10 tasks, each focusing on a different class. We retain that such a benchmark diverges from classical CL ones, as it is simplified and far from real-world scenarios. In our work, we utilize the same setting presented by the authors of [4], who split NTU RGB+D 60 into 6 tasks, each involving *multiple* classes.

## 3 Method

### 3.1 Preliminaries

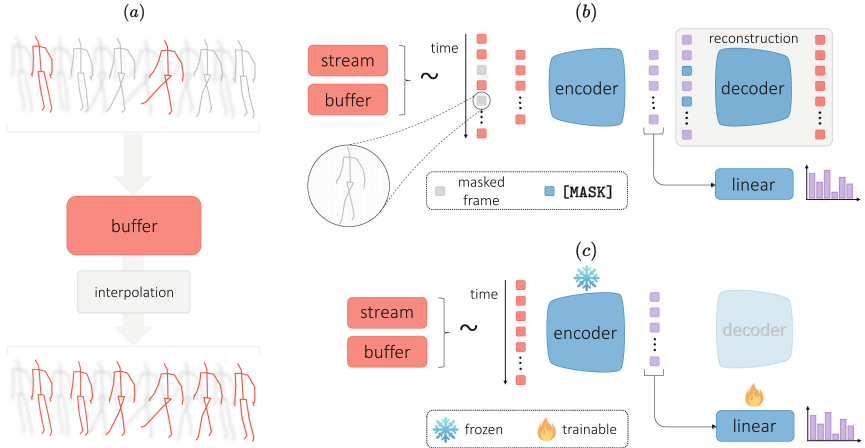
**Class-Incremental Learning.** In Class-IL, a deep model  $f(\cdot; \theta)$  parametrized by  $\theta$  is presented with a sequence of tasks  $\mathcal{T}_i$  with  $i \in \{1, \dots, T\}$ , with  $T$  denoting the number of tasks. The  $i$ -th task provides  $N_i$  data entries  $\{x_i^{(n)}, y_i^{(n)}\}_{n=1}^{N_i}$  with  $y_i^{(n)} \in \mathcal{Y}_i$ ; importantly, each task relies on a set of classes disjoint from others s.t.  $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset \iff i \neq j$ . The objective of Class-IL is to minimize the empirical risk over all tasks:

$$\mathcal{L}_{\text{Class-IL}} = \sum_{i=1}^T \mathbb{E}_{(x,y) \sim \mathcal{T}_i} [\mathcal{L}(f(x; \theta), y)], \quad (1)$$

where  $\mathcal{L}$  is the loss function (*e.g.*, the cross entropy for classification) and  $y$  is the ground truth label. Since the model observes one task at a time, tailored strategies are required to prevent catastrophic forgetting. Specifically, some rehearsal approaches [5, 33] employ an additional regularization term  $\mathcal{L}_{\mathcal{M}}$  exploiting samples stored in the memory buffer. The objective at the current task  $\mathcal{T}_c$  is:

$$\hat{\mathcal{L}}_{\text{Class-IL}} = \mathbb{E}_{(x,y) \sim \mathcal{T}_c} [\mathcal{L}(f(x; \theta), y)] + \mathcal{L}_{\mathcal{M}}. \quad (2)$$

**Spatio-Temporal Tuples Transformer (STTFormer).** We adopt as main backbone of our architecture STTFormer [31], a transformer-based model



**Fig. 1.** Figure showing the key components of CHARON. Our efficient buffer strategy is shown on the left (a). In the upper right (b), we showcase the training phase with the reconstruction regularization, while linear probing is displayed at the bottom (c). Best seen in colors.

designed for skeleton-based action recognition. It exploits self-attention to capture the cross-joint correlations across adjacent frames. Specifically, a raw skeleton sequence  $x \in \mathbb{R}^{C \times F \times V}$ , where  $C$  is the number of channels (*i.e.*, spatial coordinates),  $F$  the number of frames, and  $V$  the number of joints, is given as input to the model. This sample is divided into tuples, *i.e.*, sequences of  $n$  adjacent frames:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\lfloor F/n \rfloor}], \text{ where } \mathbf{x}_i \in \mathbb{R}^{C \times n \times V}. \quad (3)$$

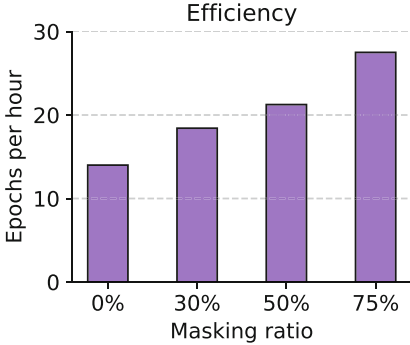
Each layer of STTFormer comprises two distinct modules, which target either *intra*- or *inter*-tuple relationships. Every element of  $\mathbf{X}$  (*i.e.*, each tuple) is first fed to a self-attention layer, which attends the joints in  $\mathbf{x}_i$ . This phase aims to model the *intra*-tuple characteristics. Then, an *inter*-tuple representation is extracted via temporal pooling.

### 3.2 CHARON

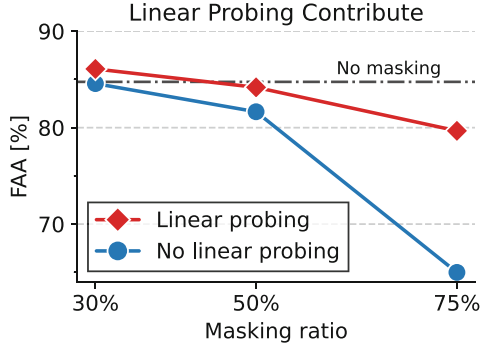
In this section, we present CHARON, which encompasses three components: *i*) a technique to populate the memory buffer, employing linear interpolation to decompress memory samples; *ii*) an efficient training phase with masked inputs; *iii*) a linear probing stage, which refines the classifier and updates the logits stored in the memory buffer. We depict these elements in Fig. 1.

**Efficient Buffer.** A raw skeleton sequence  $x \in \mathbb{R}^{C \times F \times V}$  collects the  $C$  coordinates (*e.g.*,  $xyz$  in NTU-60 and NTU-120) of  $V$  joints at  $F$  time instants. Unlike RGB video frames, skeletal data inherently reside in Euclidean space where the





**Fig. 2.** Epochs per hour at different masking ratio values.<sup>1</sup>



**Fig. 3.** Linear probing contribute on joint training with varying masking ratios.

concept of distance between points (three-dimensional joints in our case) is well-defined. Additionally, skeleton sequences often exhibit temporal redundancy [16]. In light of these peculiarities, skeletal data can be easily compressed upon need: for instance, we do so before storing a sequence into the memory buffer. Notably, the compressed sequences can also be reconstructed with minimal loss through a simple linear interpolation. In particular, even with a sampling interval of  $s = 5$  frames – *i.e.*, one kept every five, yielding a compression ratio of 80% – the reconstructions are close to the raw samples. Based on that, a greater number of instances can be stored within the same memory constraints: in other words, we can accumulate a number of samples  $s$  times larger in the buffer.

When a sample has to be extracted from the buffer for rehearsal, we reconstruct it to obtain  $F$  frames again and then treat it as a complete sample. It is noted that, since linear interpolation does not require learnable parameters, the reconstruction of temporal skeletal sequences requires low computational effort.

**Training Phase.** As we mentioned above, a transformer-based architecture founded on [31] is adopted as our backbone. We build upon it to derive an encoder-decoder framework inspired by masked autoencoders [17]. Notably, this allows us to reduce the computational effort during training, as depicted in Fig. 2. Specifically, given a sample  $x$  coming from the current task or the buffer, the first step consists of a linear projection, followed by positional encoding to inject temporal dependencies. Afterward, we feed the encoder  $e(\cdot; \theta_e)$  with a temporally masked sample  $\tilde{x} \in \mathbb{R}^{C \times \lfloor (1-\eta) \cdot F \rfloor \times V}$  obtained by dropping a random subset of frames from the input sequence, where  $\eta \in [0, 1)$  is the masking ratio.

The encoder projects the input  $\tilde{x}$  into the latent space, obtaining features  $\tilde{h} = e(\tilde{x}; \theta_e)$ . From this point, the architecture devises two branches: the first one (*recognition*) features a fully connected layer  $f(\cdot; \theta_f)$  to yield pre-softmax logits  $z = f(\tilde{h}; \theta_f)$ . The second branch (*reconstruction*) realizes the self-supervised

<sup>1</sup> Tests are performed on a single GTX 1080 Ti graphics card

regularization through a decoder module  $d(\cdot; \theta_d)$ . Specifically, given the latent feature vector  $\tilde{h}$  which has  $\lfloor (1 - \eta) \cdot F \rfloor$  tokens, the input of the decoder is formed by filling the missing ones with learnable mask vectors denoted with [MASK]. We place these vectors in the same position as the original masked ones,  $h = \text{CONCAT}(\tilde{h}, [\text{MASK}])$ . The training objective is:

$$\mathcal{L}_{\text{stream}} = \mathcal{L}_{\text{CE}}(z, y) + \gamma \cdot \|d(h; \theta_d) - x\|_2^2, \quad (4)$$

where  $\gamma$  is a hyper-parameter weighting the impact of the reconstruction loss.

To mitigate forgetting, we incorporate the objective defined in Eq. 4 into a rehearsal-based framework. Drawing inspiration from [5], we retrieve a mini-batch of samples  $x_{\mathcal{M}}$  from the memory buffer at each training step. This mini-batch includes associated predictions  $z_{\mathcal{M}}$  (*i.e.*, logits) and labels  $y_{\mathcal{M}}$ , which are added to the episodic memory along with the corresponding samples. The loss functions for these two components are:

$$\mathcal{L}_{\text{logits}} = \|f(\tilde{h}_{\mathcal{M}}; \theta_f) - z_{\mathcal{M}}\|_2^2 + \gamma \cdot \|d(h_{\mathcal{M}}; \theta_d) - x_{\mathcal{M}}\|_2^2, \quad (5)$$

$$\mathcal{L}_{\text{labels}} = \mathcal{L}_{\text{CE}}(f(\tilde{h}_{\mathcal{M}}; \theta_f), y_{\mathcal{M}}) + \gamma \cdot \|d(h_{\mathcal{M}}; \theta_d) - x_{\mathcal{M}}\|_2^2. \quad (6)$$

The mini-batch of samples  $x_{\mathcal{M}}$  undergoes the same pipeline of the input stream  $x$ , producing the latent features  $\tilde{h}_{\mathcal{M}} = e(\tilde{x}_{\mathcal{M}}; \theta_e)$  and  $h_{\mathcal{M}} = \text{CONCAT}(\tilde{h}_{\mathcal{M}}, [\text{MASK}])$ .

The final objective of this phase is:

$$\mathcal{L} = \mathcal{L}_{\text{stream}} + \alpha \cdot \mathcal{L}_{\text{logits}} + \beta \cdot \mathcal{L}_{\text{labels}}, \quad (7)$$

where  $\alpha$  and  $\beta$  are two balancing hyperparameters.

**Linear Probing.** As described above, the model is trained with partial skeleton sequences. While providing an efficient training strategy, there is a factor that could hinder the overall performance during evaluation. Indeed, we argue that the classification heads  $f(\cdot; \theta_f)$  could be subject to possible misalignment due to the different conditions we have at training (masking *on*) and test time (masking *off*). To address this issue, highlighted in Fig. 3, we devise an auxiliary linear probing stage at the end of each task, which lasts for a few epochs (*i.e.*, 10% of the number employed for the main training stage). During this phase, only the parameters of the classifier are allowed to change, while the encoder remains frozen. In doing so, we feed each full (*i.e.*, not masked) sample  $x \in \mathbb{R}^{C \times F \times V}$  to the encoder.

In formal terms, as for the main training phase, the encoder projects the input  $x$  into the latent space obtaining hidden features  $h = e(x; \theta_e)$ . The fully connected linear layer  $f(\cdot; \theta_f)$  produces then the logits  $z = f(h; \theta_f)$  to which a cross-entropy loss is finally applied. In this phase, we still employ the regularization from [5]. Thus, the resulting objective  $\mathcal{L}_{\text{lp}}$  can be written as:

$$\mathcal{L}_{\text{lp}} = \mathcal{L}_{\text{CE}}(z, y) + \alpha \cdot \|f(h_{\mathcal{M}}; \theta_f) - z_{\mathcal{M}}\|_2^2 + \beta \cdot \mathcal{L}_{\text{CE}}(f(h_{\mathcal{M}}; \theta_f), y_{\mathcal{M}}). \quad (8)$$

---

**Algorithm 1.** Training CHARON at the current task

---

**Requires:** dataset  $D_{\mathcal{T}_c}$ , parameters  $\theta$  ( $\theta_e, \theta_f, \theta_d$ ), scalars  $\alpha, \beta$  and  $\gamma$ , learning rate  $\lambda$ , masking ratio  $\eta$ , buffer  $\mathcal{M}$ .

**Main training phase:**

**for**  $(x, y)$  **in**  $D_{\mathcal{T}_c}$  **do**

$(x_{\mathcal{M}}, y_{\mathcal{M}}, z_{\mathcal{M}}) \leftarrow \text{interpolate}(\text{extract}(\mathcal{M}))$

$\tilde{x}, \tilde{x}_{\mathcal{M}} \leftarrow \text{random\_masking}(x, \eta), \text{random\_masking}(x_{\mathcal{M}}, \eta)$

$\mathcal{L} \leftarrow \text{Eqs. (4) to (7)}$

$\theta \leftarrow \theta - \lambda \cdot \nabla_{\theta} \mathcal{L}$

**end for**

**Linear probing:**

**for**  $(x, y)$  **in**  $D_{\mathcal{T}_c}$  **do**

$(x_{\mathcal{M}}, y_{\mathcal{M}}, z_{\mathcal{M}}) \leftarrow \text{interpolate}(\text{extract}(\mathcal{M}))$

$\mathcal{L} \leftarrow \text{Eqs. (8)}$

$\theta_f \leftarrow \theta_f - \lambda \cdot \nabla_{\theta_f} \mathcal{L}$

$\mathcal{M} \leftarrow \text{populate}(\mathcal{M}, (\text{uniform\_sampling}(x), z, y))$

**end for**

---

Traditional works using masked autoencoders [17, 43] typically distinguish between a pre-train phase and one of linear probing to adapt to downstream tasks. However, we argue that leading these stages separately can result in a more cumbersome approach, potentially undermining the efficiency we seek. To solve this, Eqs. (7) and (8) are computed sequentially during each task, according to the incremental setting (*i.e.*, holding only a partial amount of data, the one belonging to the current task). The complete algorithmic procedure for a single task is described in Algorithm 1.

## 4 Experimental Analysis

### 4.1 Datasets

**Split NTU-60 and Split NTU-120.** NTU is one of the most popular benchmarks for action recognition on skeletal data. Initially comprising 60 classes and 56578 samples in its original version [38], and later expanded to 120 classes and 113945 samples [28], this dataset encompasses a diverse range of actions involving up to two individuals. The data collection process involves three Kinect cameras [54], positioned with different angles w.r.t. the subject. They provide RGB videos, IR videos, depth map sequences, and 3D skeletal data. Participants of various ages have contributed to the datasets construction, ensuring its broad applicability and relevance.

We adopt the extraction process employed by [31]. As original raw sequences contain a varying number of frames, we apply bilinear interpolation to obtain fixed-length sequences  $x$  (*i.e.*, 120 frames) s.t.  $x \in \mathbb{R}^{(C=3) \times (F=120) \times (V=25) \times (B=2)}$ . The axis identified by  $B$  regards the poses of the potentially two subjects involved in the action.

To test our approach in the CL scenario, we embrace NTU-60, introduced in [4], an incremental learning benchmark derived from the standard NTU dataset. The authors of [4] divide NTU RGB+D data into 6 tasks, each defining a 10-class classification problem. We also introduce NTU-120, an extension of the previous benchmark. Such a version brings a significant additional challenge, as seen in recent offline literature [12, 14], leaving the way open for future works in the continual domain. To be as compliant as possible with the previous literature, we keep the original 6 tasks split and add another 6, each consisting of 10 classes, resulting in a 12 tasks incremental scenario. We describe in the supplementary material the exact order in which classes are split into tasks.

We report results for the cross-subject (XSub) and cross-view (XView) data modalities [38] for NTU-60, and cross-subject (XSub) and cross-setup (XSet) [28] for NTU-120.

## 4.2 Implementation Details

The custom version we adopt for STTFormer [31] reduces the width of intermediate layers to obtain a more lightweight model. We set the number of frames in each tuple  $n = 6$  as in the original paper. Following the asymmetric design proposed in [17], we employ 8 layers for the encoder and 3 for the decoder. We refer the reader to the supplementary material for further details. Additionally, we employ an  $\alpha$  of 0.3 and a  $\beta$  of 0.8 for Eqs. (7) and (8), while we use a  $\gamma$  of 0.5 in approaches using the reconstruction regularization Eqs. (4) to (6). We adopt a batch size of 16 for all our experiments with a vanilla SGD optimizer and a learning rate of 0.05. Each task of the incremental setting lasts for 30 epochs. With the same hyperparameters as above, we perform 3 epochs for the linear probing phase. Finally, concerning data augmentation, we follow the original STTFormer implementation, applying a simple random rotation to each input sample.

## 4.3 Results

For the experimental comparison, we indicate with Joint Training (JT) the upper bound of our approach. It consists of training the model on the unified dataset (*i.e.*, without splitting it into tasks). For the lower bound, we adopt an incremental training approach that does not employ tailored techniques against catastrophic forgetting. We refer to it as Fine Tuning (FT).

In Table 1 we report the results for buffer sizes  $\mathcal{M}_{size}$  of dimensions 500 and 2000. Following other works [4, 5, 26], we measure the recognition performance in terms of Final Average Accuracy (FAA), defined as:

$$\text{FAA} = \frac{1}{T} \sum_{i=1}^T a_{\mathcal{T}_i}, \quad (9)$$

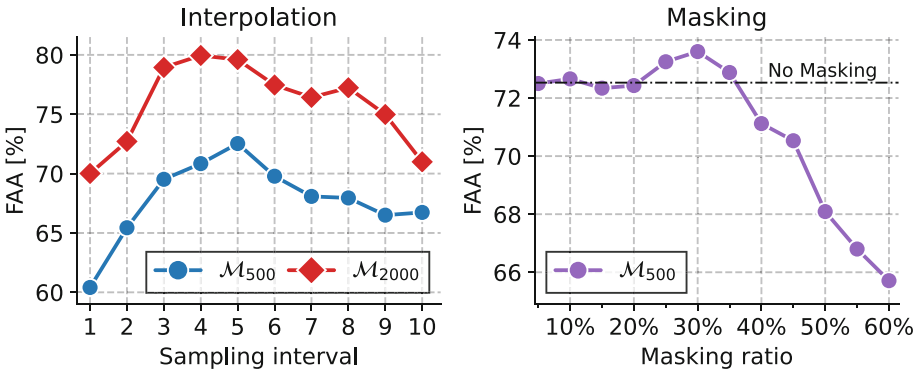
where  $a_{\mathcal{T}_i}$  is the accuracy of the  $i$ -th task after the model has seen all  $T$  of them. Additionally, we repeat each experiment three times, thus reporting the mean and standard deviation of the FAA.

**Table 1.** FAA (%) results on NTU-60 and NTU-120. For **CHARON**, we report the results with a masking ratio equal to 30%. We highlight in green the gains achieved by our approach w.r.t. the best-competing method.

Method	XView		XSub		XSet		XSub	
<b>FT</b>	16.05 $\pm$ 0.07		15.64 $\pm$ 0.05		7.19 $\pm$ 0.06		6.97 $\pm$ 0.23	
<b>JT</b>	84.75 $\pm$ 0.02		77.32 $\pm$ 0.54		71.18 $\pm$ 1.07		70.15 $\pm$ 0.98	
$\mathcal{M}_{size}$	500	2000	500	2000	500	2000	500	2000
<b>iCaRL</b>	51.54 $\pm$ 1.3	53.41 $\pm$ 1.1	47.12 $\pm$ 1.4	50.69 $\pm$ 1.2	32.91 $\pm$ 0.9	34.74 $\pm$ 0.7	33.03 $\pm$ 1.3	36.68 $\pm$ 1.0
<b>Else-Net</b>	40.81 $\pm$ 0.8	59.10 $\pm$ 0.2	39.72 $\pm$ 0.4	57.00 $\pm$ 1.0	19.37 $\pm$ 0.6	33.52 $\pm$ 0.6	18.43 $\pm$ 0.7	33.95 $\pm$ 0.3
<b>ER</b>	51.00 $\pm$ 1.6	68.27 $\pm$ 0.1	45.80 $\pm$ 0.5	62.74 $\pm$ 1.9	26.35 $\pm$ 1.1	43.12 $\pm$ 0.4	26.19 $\pm$ 1.7	45.06 $\pm$ 0.7
<b>DER</b>	51.36 $\pm$ 0.9	66.74 $\pm$ 0.1	49.97 $\pm$ 1.9	63.48 $\pm$ 1.3	27.83 $\pm$ 1.7	40.19 $\pm$ 0.9	30.10 $\pm$ 1.5	36.10 $\pm$ 1.8
<b>DER++</b>	60.41 $\pm$ 0.5	73.09 $\pm$ 1.3	57.22 $\pm$ 1.0	67.64 $\pm$ 1.6	34.27 $\pm$ 1.4	50.06 $\pm$ 0.6	36.29 $\pm$ 0.3	49.81 $\pm$ 0.8
<b>CHARON</b>	<b>73.60<math>\pm</math>0.3</b>	<b>77.77<math>\pm</math>0.2</b>	<b>68.30<math>\pm</math>0.6</b>	<b>72.70<math>\pm</math>0.2</b>	<b>52.19<math>\pm</math>0.6</b>	<b>61.63<math>\pm</math>0.1</b>	<b>48.64<math>\pm</math>0.0</b>	<b>59.23<math>\pm</math>0.4</b>
	<b>+13.19</b>	<b>+4.68</b>	<b>+11.08</b>	<b>+5.06</b>	<b>+17.92</b>	<b>+11.57</b>	<b>+12.35</b>	<b>+9.42</b>

As outlined by Table 1, the main competitor of this work, Else-Net [26], did not achieve performance comparable to those of the setting proposed by its authors, which devises a massive pre-training phase. Therefore, we can conclude that such a method suffers when trained from scratch.

Furthermore, even classical replay methods such as iCaRL [32], ER [33] and DER(++) [5] outperform Else-Net. CHARON reveals to be SOTA in the Class-IL skeleton-based action recognition domain, across both NTU-60 and NTU-120. In particular, this holds when employing a *masking ratio* of 30%; for higher percentages, we observe a decrease in performance, as discussed in the following. Significantly, the most substantial improvement is observed with a buffer size of 500 (surpassing the second-best, *i.e.*, DER++, when using a buffer size of 2000). This highlights the pivotal role of the sample quantity in the efficacy of replay methods. Consequently, it underscores the importance of researching techniques to increase sample numbers within a fixed buffer size.



**Fig. 4.** (left) FAA for the DER++ baseline employing different values of the sampling interval  $s$ . (right) FAA obtained by CHARON as the masking ratio varies.

**On the Sampling Interval.** To further evaluate the effectiveness of our buffer strategy, we conduct a comparative study on varying *sampling interval*  $s$  (which we recall indicates the step length in the uniform sampling procedure). Given  $s \in \mathbb{N}^+$ , we obtain the *compression ratio* as:

$$\text{compression ratio} = \frac{s-1}{s} \cdot 100. \quad (10)$$

We report in Fig. 4 (*left*) the FAA at varying sampling interval  $s$  for both the buffer sizes tested. For each tested sampling interval, we scale the buffer size accordingly (as documented in Fig. 3.2). For instance, when  $s = 10$ , a memory with a nominal capacity of 500 examples could hence contain at most  $s \cdot 500 = 5000$  (compressed) examples. As can be appreciated, the sampling interval  $s = 5$  (*i.e.*, 80% of *compression*) yields the best results in terms of final accuracy. Namely, when sampling one skeletal pose every five frames, the memory buffer attains the best compromise between sample *fidelity* (which can be achieved with lower sampling intervals) and sample *diversity* (*i.e.*, higher intervals). Moreover, we note that the presence of a prior compression phase ( $s > 1$ ) brings a stable and remarkable gain w.r.t. the standard replaying paradigm ( $s = 1 \rightarrow$  no compression at all). Such a result shows the crucial role of the trade-off between the quality and quantity of samples.

**On the Masking Ratio.** We herein assess the impact of the *masking ratio*, which indicates the number of frames discarded before feeding the input sequence to the model. The results are illustrated in Fig. 4 (*right*) and reveal an increase in performance up to a value of 30%. For higher masking ratios, performance begins to decline, despite the notable efficiency gains (see Fig. 2). In quantitative terms, even with 50% of masking, CHARON achieves an acceptable final average accuracy of around 68%, while it decreases to  $\approx 66\%$  with a masking ratio equal to 60%. Interestingly, both of these results are still higher than those of DER++, the second-best method reported in Tab. 1.

#### 4.4 Ablations

We herein report the ablative studies; all the experiments are performed on the XView modality of NTU-60.

**On the importance of the Reconstruction-Based Objective.** Our approach not only seeks good classification capabilities but also devises an auxiliary reconstruction term targeting the entire input sequence. To shed further light on the effects of such an auxiliary objective, we provide an ablative experiment in which we discard both the decoder module and the subsequent reconstruction loss. In doing so, we still apply random masking (testing two ratios equal to 30% and 60%) and linear probing at the end of each task.

**Table 2.** Impact of the reconstruction loss at different masking ratios.

	Masking ratio	
	30%	60%
w/o recon. loss	70.61	61.59
CHARON	<b>73.60</b>	<b>65.72</b>

**Table 3.** Ablative outcomes about sampling strategy and masking position.

Strategy	Position	
	<i>pre</i>	<i>post</i>
<i>Deterministic</i>	72.08	72.43
<i>Random</i>	71.89	<b>73.60</b>

The results of these ablative studies are reported in Tab. 2: remarkably, CHARON experiences a significant performance drop when removing the decoder and the reconstruction loss, especially for the higher masking ratio of 60%. We consider such a finding as noteworthy, as it highlights the importance of auxiliary learning techniques when leveraging higher compression ratios to pursue efficiency.

**Masking Strategy and Positioning.** Our approach adopts a masking strategy that builds upon random guessing to drop frames, thus following most of the literature dealing with masked autoencoders. Herein, we want to compare our approach with a deterministic strategy, that drops one frame every  $k$ . We also assess different possible positions to introduce the masking operation. Specifically, *post* indicates that masking is placed after splitting the sequence into tuples (see Sect. 3.1), as carried out by our approach. Results for the combinations of these two alternatives are reported in Tab. 3: as can be observed, the random strategy with post-hoc masking emerges as the best configuration.

## 5 Conclusions

Skeleton-based action recognition is a relevant task in modern human-centric Artificial Intelligence. We addressed such a long-standing computer vision task from the perspective of incremental learning, thus enabling those applications (*e.g.*, sports analysis, rehabilitative healthcare) where the set of actions to be recognized may change over time. Differently from existing proposals dealing with action recognition, our work appoints *efficiency* as a crucial aspect of an ideal incremental learner.

Our method, named CHARON, could be considered a step forward, as it achieves state-of-the-art performance with a remarkable reduction of the computational footprint (in terms of both memory and training time). In a few words, these capabilities derive from a proper application of input sub-sampling and random masking. Importantly, our experiments show that the addition of a reconstruction-based auxiliary objective grants further robustness in the presence of higher masking ratios, thus encompassing settings demanding efficiency. In future studies, we are going to deepen the concepts discussed in this paper, to apply our proposal even in the case of extreme masking (*e.g.*, up to 95%).

**Acknowledgements.** Andriy Sorokin was supported by Marie Skłodowska-Curie Action Horizon 2020 (Grant agreement No. 955778) for the project “Personalized Robotics as Service Oriented Applications” (“PERSEO”). Additionally, the research activities of Angelo Porrello have been partially supported by the Department of Engineering “Enzo Ferrari” through the program FAR\_2023\_DIP – CUP E93C23000280005.

## References

1. Arani, E., Sarfraz, F., Zonooz, B.: Learning fast, learning slow: a general continual learning method based on complementary learning system. In: International Conference on Learning Representations (2022)
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022)
3. Bonato, J., Pelosin, F., Sabetta, L., Nicolosi, A.: Mind: multi-task incremental network distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2024)
4. Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., Calderara, S.: Class-incremental continual learning into the extended der-verse. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
5. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. *Adv. Neural Inform. Process. Syst.* (2020)
6. Castagnolo, G., Spampinato, C., Rundo, F., Giordano, D., Palazzo, S.: A baseline on continual learning methods for video action recognition. *arXiv preprint [arXiv:2304.10335](https://arxiv.org/abs/2304.10335)* (2023)
7. Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., Hu, W.: Channel-wise topology refinement graph convolution for skeleton-based action recognition. In: IEEE International Conference on Computer Vision (2021)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS Workshop on Deep Learning (2014)
9. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
10. Du, Y., Fu, Y., Wang, L.: Skeleton based action recognition with convolutional neural network. In: Proceedings of the Asian Conference on Computer Vision (2015)
11. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015)
12. Duan, H., Wang, J., Chen, K., Lin, D.: Dg-stgcn: dynamic spatial-temporal modeling for skeleton-based action recognition. *arXiv preprint [arXiv:2210.05895](https://arxiv.org/abs/2210.05895)* (2022)
13. Duan, H., Wang, J., Chen, K., Lin, D.: Pyskl: towards good practices for skeleton action recognition. In: ACM International Conference on Multimedia (2022)
14. Duan, H., Zhao, Y., Chen, K., Lin, D., Dai, B.: Revisiting skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
15. Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M.: Pstnet: point spatio-temporal convolution on point cloud sequences. In: International Conference on Learning Representations (2021)



16. González-Aparicio, M.T., García, R., Brugos, J., Pañeda, X.G., Melendi, D., Cabrero, S.: Measuring temporal redundancy in sequences of video requests in a news-on-demand service. *Telematics and Informatics* (2014)
17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2022)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* (1997)
19. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning* (2015)
20. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017)
21. Kim, T.S., Reiter, A.: Interpretable 3d human action analysis with temporal convolutional networks. In: *IEEE International Conference on Computer Vision and Pattern Recognition Workshops* (2017)
22. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *Proc. National Acad. Sci.* (2017)
23. Kong, Y., Fu, Y.: Human action recognition and prediction: a survey. *Inter. J. Comput. Vis.* (2022)
24. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* (2011)
25. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: *IEEE International Conference on Multimedia and Expo Workshops* (2017)
26. Li, T., Ke, Q., Rahmani, H., Ho, R.E., Ding, H., Liu, J.: Else-net: elastic semantic network for continual action recognition from skeleton data. In: *IEEE International Conference on Computer Vision* (2021)
27. Lin, W., Sun, M.T., Poovandran, R., Zhang, Z.: Human activity recognition for video surveillance. In: *IEEE International Symposium on Circuits and systems (ISCAS)* (2008)
28. Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y., Kot, A.C.: Ntu rgb+ d 120: a large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019)
29. Panariello, A., Porrello, A., Calderara, S., Cucchiara, R.: Consistency based self-supervised learning for temporal anomaly localization. In: *European Conference on Computer Vision Workshops* (2022)
30. Park, J., Kang, M., Han, B.: Class-incremental learning for action recognition in videos. In: *IEEE International Conference on Computer Vision* (2021)
31. Qiu, H., Hou, B., Ren, B., Zhang, X.: Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint [arXiv:2201.02849](https://arxiv.org/abs/2201.02849)* (2022)
32. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: incremental classifier and representation learning. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2017)
33. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., , Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. In: *International Conference on Learning Representations* (2019)
34. Robins, A.: Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science* (1995)
35. Rusu, A.A., et al.: Progressive neural networks. *arXiv preprint [arXiv:1606.04671](https://arxiv.org/abs/1606.04671)* (2016)

36. Sanchez-Caballero, A., Fuentes-Jimenez, D., Losada-Gutiérrez, C.: Exploiting the convlstm: Human action recognition using raw depth video-based recurrent neural networks. arXiv preprint [arXiv:2006.07744](https://arxiv.org/abs/2006.07744) (2020)
37. Schwarz, J., et al.: Progress & compress: a scalable framework for continual learning. In: International Conference on Machine Learning (2018)
38. Shahroudy, A., Liu, J., Ng, T., Wang, G.: Ntu rgb+d: a large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
39. Shao, D., Zhao, Y., Dai, B., Lin, D.: Finegym: a hierarchical video dataset for fine-grained action understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
40. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In: Proceedings of the Asian Conference on Computer Vision (2020)
41. Smith, J.S., et al.: Coda-prompt: continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2023)
42. Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human action recognition from various data modalities: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
43. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: masked autoencoders are data-efficient learners for self-supervised video pre-training. *Adv. Neural Inform. Process. Syst.* (2022)
44. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint [arXiv:1904.07734](https://arxiv.org/abs/1904.07734) (2019)
45. Villa, A., et al.: Pivot: prompting for video continual learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2023)
46. Wang, Z., et al.: Learning to prompt for continual learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (2022)
47. Wu, W., Hua, Y., Zheng, C., Wu, S., Chen, C., Lu, A.: Skeletonmae: spatial-temporal masked autoencoders for self-supervised skeleton action recognition. In: IEEE International Conference on Multimedia and Expo Workshops (2023)
48. Xin, W., Liu, R., Liu, Y., Chen, Y., Yu, W., Miao, Q.: Transformer for skeleton-based action recognition: a review of recent advances. *Neurocomputing* (2023)
49. Yan, H., Liu, Y., Wei, Y., Li, Z., Li, G., Lin, L.: Skeletonmae: graph-based masked autoencoder for skeleton sequence pre-training. In: IEEE International Conference on Computer Vision (2023)
50. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
51. Yin, J., Han, J., Wang, C., Zhang, B., Zeng, X.: A skeleton-based action recognition system for medical condition detection. In: IEEE Biomedical Circuits and Systems Conference (BioCAS) (2019)
52. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning (2017)
53. Zhang, S., et al.: Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Trans. Multimedia* (2018)
54. Zhang, Z.: Microsoft kinect sensor and its effect. *IEEE MultiMedia* (2012)



# Text-Driven Prototype Learning for Few-Shot Class-Incremental Learning

Seongbeom Park<sup>1</sup>, Haeji Jung<sup>1</sup>, Daewon Chae<sup>1</sup>, Hyunju Yun<sup>1</sup>, Sungyoon Kim<sup>2</sup>,  
Suhong Moon<sup>2</sup>, Jinkyu Kim<sup>1</sup>, and Seunghyun Park<sup>3</sup>(✉)

<sup>1</sup> Korea University, Seoul 02841, Korea

<sup>2</sup> University of California, Berkeley, CA 94720, USA

<sup>3</sup> NAVER Cloud AI, Seongnam-si, Gyeonggi-do 13529, Korea  
seung.park@navercorp.com

**Abstract.** Few-shot class-incremental learning (FSCIL) aims to learn generalizable representations with large amounts of initial data and incrementally adapt to new classes with limited data (*i.e.*, few-shot). Recently, prototype-based approaches have shown notably improved performance. However, there still remain challenges – their performances often degrade when newly added classes have high similarity with previously seen classes, causing prototypes to be indistinguishable. In this work, we advocate for leveraging textual semantics to learn class-representative and class-distinguishable prototypes, retaining semantic relations between classes. We utilize angular margin loss to leverage textual semantics effectively, encouraging the model to have intra-class compactness and inter-class discrepancies in the embedding space. Our experiments with three public benchmarks (CUB200, CIFAR100, and miniImageNet) show that our proposed method generally matches or outperforms the current state-of-the-art approaches. To further demonstrate the effectiveness of using texts in the FSCIL task, we newly collect visually descriptive and class-discriminative descriptions built upon two widely-used FSCIL benchmarks: CIFAR100-Text and miniImageNet-Text.

**Keywords:** Few-Shot Class-Incremental Learning · Text-Driven Prototype

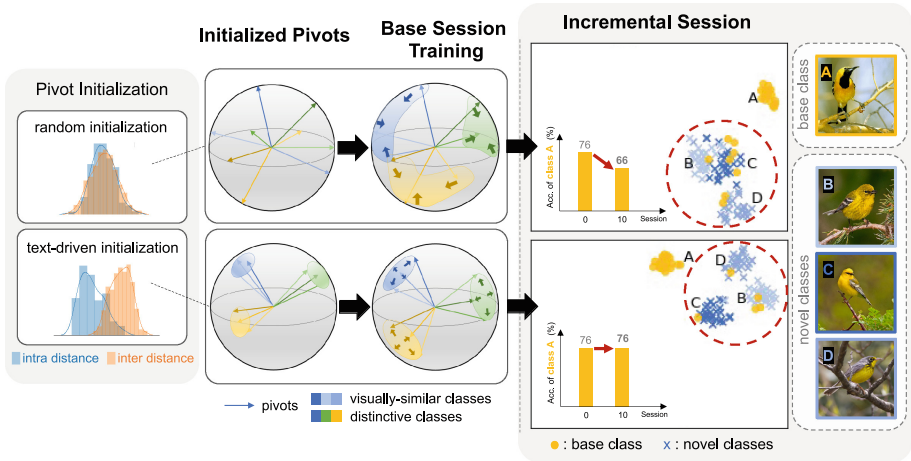
## 1 Introduction

The field of few-shot class-incremental learning (FSCIL) [26] has attracted significant interest for its relevance and promise in practical scenarios, by integrating two critical challenges: (i) class-incremental learning, which requires a model to incorporate new classes into its existing knowledge base without suffering from

S. Park and H. Jung—These authors contributed equally to this work.

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_2](https://doi.org/10.1007/978-3-031-78189-6_2).



**Fig. 1.** Our model leverages textual semantics to learn class-representative and class-distinguishable prototypes. Unlike randomly-initialized prototypes, text-driven prototypes produce an embedding space that is effective in retaining semantic relations between classes, thus eventually improving overall FSCIL performance. We provide t-SNE visualizations of learned features from different but visually-similar classes where text-driven features are more dispersed from other classes, preventing the loss of accuracy (for the base class A) in the incremental sessions.

catastrophic forgetting [19] (*i.e.*, without losing previously learned knowledge). (ii) Few-shot learning involves training the model with limited data samples, ensuring its ability to adapt to novel, unseen data without overfitting to these new classes. Tackling FSCIL commonly consists of two steps: (1) Pre-training with base classes, where a model utilizes large amounts of data, and (2) Few-shot adaptation to additional classes while preventing performance degradation on previously seen classes at the same time.

Various methods have been introduced to address the FSCIL problem [27], and recent successes suggest that prototype-based learning methods can significantly improve overall performance [2, 10, 21, 36, 40] in terms of classification accuracy and performance dropping rate. In these methods, a model is trained to learn per-class representative pivots (or prototypes) for a given dataset, classifying data points by comparing similarities with each pivot later. Thus, a key component of the prototype-based learning methods is learning class-distinguishable prototypes, which can generalize to unseen new classes. Various approaches have been applied to learn better prototypes, including regularizer of per-class embedding distributions [40], dynamic relation projection [43], prototype clustering approach [2], generating quasi-orthogonal prototypes [10], incorporating an angular penalty loss [21], and human cognition-inspired prototypes [36].

However, there still remain challenges: (i) model’s performance often largely degrades when newly added classes have high similarity with previously seen classes, causing prototypes to be close to each other and no longer distinguish-

able. Other challenges may include (ii) data imbalance between data-rich base classes and newly added small amounts of classes. As shown in Fig. 1, to address these issues, we advocate for leveraging semantics from textual modality to learn class-representative prototypes. We argue that using text in the FSCIL task is advantageous for the following reasons: (1) inter-class relations can be easily captured by the textual modality, and (2) text-driven prototypes may reduce the semantic gap between the few-shot class prototypes and the real data distribution, improving model’s ability to generalize. We use the generalized angular margin penalty-based loss to leverage textual semantics effectively while maintaining intra-class compactness and inter-class discrepancy in the embedding space.

Further, an FSCIL dataset, consisting of a pair of texts and images, is needed to demonstrate the effectiveness of using texts. Although the CUB200 [29] dataset has a collection of natural language descriptions [23], other widely-used FSCIL benchmarks (*e.g.*, CIFAR100 [14] and miniImageNet [28]) do not. Thus, we create new datasets, called CIFAR100-Text and miniImageNet-Text, by collecting textual descriptions, which are visually descriptive and class-discriminative. Inspired by recent work [20, 35], we utilize a large language model (LLM), such as GPT-3 [1], to collect such textual descriptions instead of recruiting human annotators. Our collection process consists of three steps: (1) Candidate Descriptive Words Generation, (2) Filtering of Visually-Non-Matched Descriptions, and (3) Complete Sentence Generation.

We demonstrate the effectiveness of our proposed method with three widely-used benchmarks: CUB200 [29], CIFAR100 [14], and miniImageNet [28]. Our experiments demonstrate that our proposed method generally matches or outperforms the current state-of-the-art approaches, confirming our model’s ability to learn class-discriminative semantic-distilled prototypes from textual modality. Our contributions are summarized as follows:

- We propose a novel prototype-based FSCIL method where prototypes are initialized and learned along with textual semantics, retaining semantic relations between different classes.
- We demonstrate the effectiveness of our method on three widely-used public benchmarks: CUB200, CIFAR100, and miniImageNet. Our model generally matches or outperforms the other state-of-the-art approaches in terms of the average accuracy and performance dropping rate.
- We create new datasets, called CIFAR100-Text and miniImageNet-Text, for leveraging textual modality in the FSCIL task by collecting class-discriminative and visually-descriptive sentences with a GPT-3-based large language model (LLM).

## 2 Related Works

**Few-Shot Class-Incremental Learning.** Few-shot class-incremental learning (FSCIL) [26] aims to incrementally train a model with new class sets that

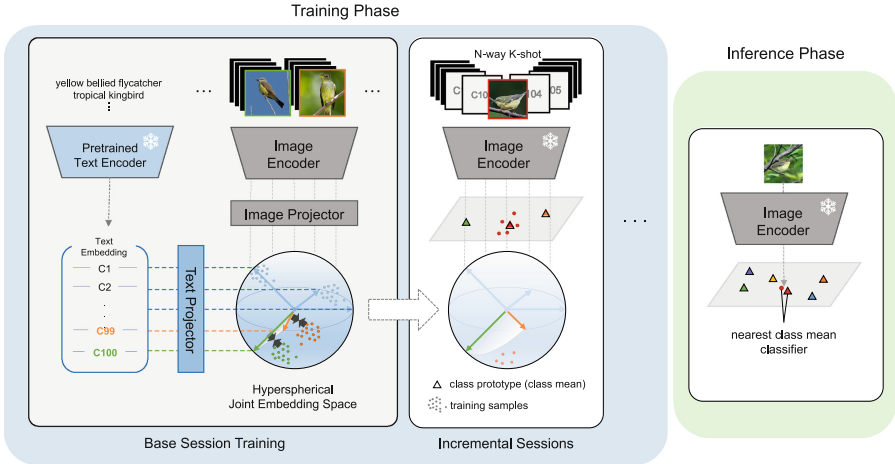
have only a few samples, while preserving knowledge of previously learned classes. Contrary to the standard class-incremental learning framework, FSCIL requires the model to adjust to these new classes using a limited number of examples per class (*e.g.*, 5 or 10), posing an additional challenge of adaptation alongside the risk of catastrophic forgetting. Initial research in FSCIL focused on updating the model’s backbone in each session [8, 26], but such updates during incremental sessions tend to cause overfitting due to the small sample size of new classes (*i.e.*, severe data imbalance between base and incremental session classes).

To address this, recent approaches have utilized prototype-based learning [10, 21, 34, 37, 41, 43, 44]. These approaches first optimize the backbone with a substantial amount of data during the base session, and then freeze the backbone in subsequent incremental sessions to derive new class prototypes by averaging the embeddings of each class. Prototype updates are facilitated through various methods: a meta-learning mechanism in the base session [37, 43], adjustments to projection layers [10, 34], or the use of virtual or augmented classes/samples to create a feature space with closely packed intra-class embeddings and additional space for accommodating new classes [21, 41]. For example, Peng *et al.* [21] utilize a margin-based loss to achieve such a space, while Zhou *et al.* [41] propose a bimodal distribution anticipating the inclusion of extra classes. These recent studies have yielded encouraging results, highlighting the importance of crafting an effective embedding space early on and preparing it for future classes with limited samples as a critical success factor in FSCIL. Our approach aligns with this line of works, integrating textual semantics to enhance the learning of the embedding space.

**Few-Shot Class-Incremental Learning with Textual Semantics.** While incorporating textual representation has been proven effective in few-shot learning (FSL) contexts demonstrated by several studies [3, 11, 22, 32], its application in FSCIL has been limited. Given the close relationship between FSCIL and FSL methodologies (both requiring adaptation to new classes with only few samples), it is worthwhile to explore textual integration within FSCIL.

Cheraghian *et al.* [4] made an initial attempt to map visual features to a textual semantic space, using the distance between these modalities for knowledge distillation from previous sessions. Subsequent work [5] projected both visual and textual semantic features into aligned subspaces, combining these projections to calculate relation scores for the final verdict. These approaches revealed how textual semantics could mitigate forgetting and address the imbalance between prior and current classes, thus alleviating the performance drop in incremental sessions. Nonetheless, these methods introduce additional computation and memory demands in incremental sessions by updating modules across both modalities and requiring prototypes and text embeddings for all previous classes, which would be infeasible due to the limited number of new class samples.

Therefore, our work shifts focus towards crafting an embedding space that is primed for future class integration, achieving a durable visual representation without necessitating auxiliary modules in incremental sessions. To this end,



**Fig. 2.** An overview of our proposed text-driven prototype learning approach for few-shot class-incremental learning. Our training consists of two main parts: a base session (with large amounts of data) and incremental sessions (with limited amounts of data). In the base session, we compute text-driven pivots and image features, which are optimized through angular margin penalty-based loss so that image features are compactly pulled together around the corresponding textual pivot (see leftmost column). In the incremental sessions, new prototypes are generated by taking an average over image features with a frozen backbone. Lastly, in the inference phase, the nearest class mean classifier is used for the final verdict.

we leverage textual semantics solely during the base session to develop a well-constructed embedding space, foregoing the textual encoder in subsequent incremental sessions. Our method diverges from prior efforts by not seeking to learn mappings between modalities but to learn a generalizable *visual space* enriched by textual semantics.

### 3 Method

**Problem Definition.** FSCIL presents a challenge that unfolds in two distinct stages: (i) a base session with a large amount of data, and (ii) incremental sessions with few-shot training data. In an  $m$ -step scenario, a relatively large training dataset,  $\mathcal{D}^0$ , is provided for the base session, followed by  $m$  training datasets comprising of few-shot samples for the incremental sessions,  $\mathcal{D}^1, \dots, \mathcal{D}^m$ . The few-shot samples in these incremental sessions are structured in an  $N$ -way  $K$ -shot format, where  $N$  denotes the number of classes, and  $K$  indicates the number of samples per class. Training data from each session have a corresponding label space  $\mathcal{Y}^i$  for  $i \in [0, m]$ , and these spaces are mutually exclusive across sessions; *i.e.*,  $\mathcal{Y}^i \cap \mathcal{Y}^j = \emptyset$  for  $i \neq j$ . Furthermore, training data from former sessions are not available in subsequent incremental sessions, posing a significant challenge for the model to retain knowledge acquired from earlier sessions. This retention



is crucial as the model is evaluated against the cumulative label spaces from all previous sessions, underscoring the importance of preserving prior learning.

**Overview.** As shown in Fig. 2, our model follows the standard FSCIL prototype learning strategy employed in various existing models. This strategy comprises two main steps: (1) learning a generalizable feature extractor during the base session with ample training data  $\mathcal{D}^0$ , and (2) generating new prototypes for each new class during the incremental sessions with limited training data  $\mathcal{D}^i$  for  $i \in [1, m]$ . In step (2), given a set of examples  $(\mathbf{x}_c, \mathbf{y}_c) \in \mathcal{D}^i$  for a new class  $c$ , the class prototype  $p_c$  is typically calculated by aggregating features; *i.e.*,  $p_c = \frac{1}{K} \sum f_\theta(\mathbf{x}_c)$ , where  $f_\theta$  is a feature extractor parameterized by  $\theta$ , and  $K$  represents the cardinality of a set of  $K$ -shot examples.

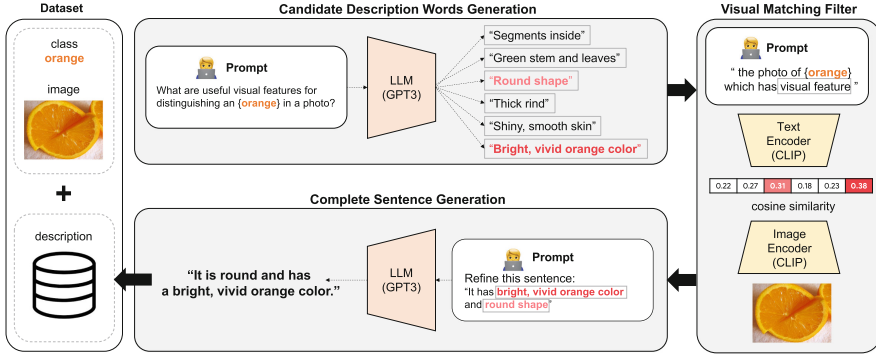
**Learning Class-Discriminative Prototypes.** Given these prototypes, new data points are classified by measuring the pairwise similarity between the data points and the prototypes, with the model outputting the nearest class (of the smallest cosine distance) as the final output. Thus, learning class-discriminative prototypes is a crucial aspect of the FSCIL model. Various approaches have been applied to enhance the distinctiveness of these prototypes [2, 10, 21, 36, 40, 43], and our work aligns with this stream of efforts to learn class-distinguishable prototypes, aiming to ensure minimal similarities between old and new prototypes and to improve overall classification performance across all sessions.

A primary difference of our work from existing efforts is the incorporation of textual modality in prototype learning, wherein textual descriptions are utilized to guide the model in generating prototypes. This approach offers advantages for the following reasons: (i) the textual modality can easily capture inter-class relations, encompassing both new and previously learned classes; and (ii) the potential of text-driven prototypes to bridge the semantic gap between the few-shot class prototypes and the actual data distribution. In the following section, we detail the application of text modality in prototype learning.

**Learning Text-Driven Prototypes.** As shown in Fig. 2, our model consists of two main components: (i) a visual encoder  $f_{\mathcal{I}}$  and (ii) a textual encoder  $f_{\mathcal{T}}$ . Given more than one natural language descriptions for a specific class, *e.g.*, “this bird has a long pointed bill with a white belly and a black crown”, we obtain sentence embeddings by utilizing a textual encoder  $f_{\mathcal{T}}$ . Note that we use a pre-trained CLIP [22]-based model as our textual encoder, which is kept frozen instead of training it from scratch. Formally, we define  $\mathbf{t}_c$  as a set of sentence embeddings for class  $c$ , *i.e.*,  $\mathbf{t}_c = \{\mathbf{t}_c^1, \mathbf{t}_c^2, \dots\}$ , where  $\mathbf{t}_c^i \in \mathbb{R}^{d_t}$  represents the  $i$ -th description embedding of class  $c$ . Subsequently, we compute text-driven pivots  $\bar{\mathbf{t}}_c$  for each class, *i.e.*,  $\bar{\mathbf{t}}_c = f_{\text{proj}, \mathcal{T}}(1/|\mathbf{t}_c| \sum_i \mathbf{t}_c^i)$ , by leveraging an MLP-based projection layer  $f_{\text{proj}, \mathcal{T}}$  to align the embedding space of both the textual and visual modalities.

Following recent work by Peng *et al.* [21], we explore the use of angular margin loss to enhance (i) intra-class compactness and (ii) inter-class discrepancy in the text-driven embedding space. Formally, given an input image  $\mathbf{x}_i$ , we compute





**Fig. 3.** An overview of our textual description collection process based on a GPT-3 [1] model. Our data collection process consists of three steps: (1) Candidate Description Words Generation, (2) Filtering of Visually-Non-Matched Textual Features, and (3) Complete Sentence Generation.

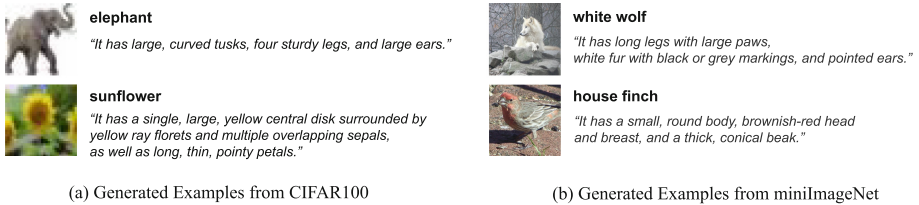
the  $d_v$ -dimensional projected visual feature  $\mathbf{v}_i \in \mathbb{R}^{d_v}$ , *i.e.*,  $\mathbf{v}_i = f_{\text{proj}, \mathcal{I}}(f_{\mathcal{I}}(\mathbf{x}_i))$ . Logits are computed by taking the dot product of the feature  $\mathbf{v}_i$  and text-driven pivots, *e.g.*,  $\|\mathbf{v}_i\| \|\bar{\mathbf{t}}_{y_i}\| \cos(\theta_{y_i})$ , where  $y_i$  denotes the ground-truth class of the input image, and  $\theta_{y_i}$  is the angle between the feature  $\mathbf{v}_i$  and the text-driven pivot  $\bar{\mathbf{t}}_{y_i}$ . By normalizing the magnitudes of the features and pivots (*i.e.*,  $\|\mathbf{v}_i\| = \|\bar{\mathbf{t}}_{y_i}\| = 1$ ), the logits are simply calculated based on cosine similarities. Based on this, we train our model by minimizing the following generalized angular margin penalty-based loss [7, 16, 30], which is defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log\left(\frac{e^{s(\cos(m_1\theta_{y_i} + m_2) - m_3)}}{e^{s(\cos(m_1\theta_{y_i} + m_2) - m_3)} + \sum_{j \neq y_i} e^{s(\cos(\theta_j))}}\right) \quad (1)$$

where  $m_1$ ,  $m_2$ ,  $m_3$ , and  $s$  are the margin penalty parameters, and  $N$  is the number of samples. We performed a grid search to find the best set of these hyperparameters, and set  $(m_1, m_2, m_3, s)$  as  $(1, 0, 0.4, 30)$ . After training, our per-class prototype  $p_c$  is computed by aggregating all visual features of a given class  $c$ :  $p_c = \frac{1}{|\{i|y_i=c\}|} \sum_{i:y_i=c} f_{\mathcal{I}}(\mathbf{x}_i)$ .

## 4 CIFAR100-Text and MiniImageNet-Text Datasets

In this work, we advocate for leveraging text modality for learning text-driven prototypes. To support this idea, an FSCIL dataset is needed that contains pairs of texts and images. However, this may be challenging because such texts should be visually descriptive and class-discriminative. To the best of our knowledge, the CUB200 [29] dataset has a collection of natural language descriptions that are suitable for class-incremental learning [23], but other datasets widely used in the CIL task rarely provide such natural language supervision. Therefore, to evaluate the effectiveness of cross-modal supervision on other widely-used datasets such



**Fig. 4.** Examples of generated descriptions for (a) CIFAR100 [14] and (b) miniImageNet [28]. Class names are in **bold** and the generated description for each image is in *italic*. More examples are provided in the supplemental material.

as CIFAR100 [14] and miniImageNet [28], we create a dataset called CIFAR100-Text and miniImageNet-Text by collecting textual descriptions.

**LLM-Based Textual Descriptions.** Inspired by [20,35], we collect textual descriptions for the FSCIL task based on a large language model (LLM) such as GPT-3 [1]. Specifically, we exploit two pre-trained models: GPT-3 [1] text-davinci-003 and CLIP model (with image and text encoders) [22] trained with ViT-B/32 Transformer as image encoder. As shown in Fig. 3, our data creation process follows three steps: (1) we use an LLM (*i.e.*, GPT-3) to collect candidate textual descriptions. For example, we query with a question like “What are useful visual features for distinguishing a horse in a photo?” and we collect answers from an LLM (*e.g.*, hooves, long mane, or broad flat head). (2) To ensure such descriptions align visually well with images, we use a pre-trained CLIP model, computing cosine similarity between images against textual features. We filter out candidate descriptions with smaller similarities than a user-defined threshold value. (3) Lastly, based on the remaining descriptions, we create a simple sentence, *e.g.*, “it has long mane and tail and small pointed ear,” followed by refinement with a text prompt such as “refine below sentences.” In Fig. 4, we provide example descriptions for CIFAR100 and miniImageNet, respectively.

**Candidate Description Words Generation.** To generate a textual description for each train image, we first obtain some candidate visual features to be integrated within a descriptive sentence. Concretely, we query the LLM to provide a selection of features that characterize the class of the given image. Following Menon *et al.* [20] and Yang *et al.* [35] which relies on the prompt design instructions provided by OpenAI, we prompt the LLM with the following form:

```
Q: What are useful visual features for distinguishing
    {a|an} {class} in a photo?
A: There are useful visual features to tell there is
    {a|an} {class} in a photo:
    - <visual feature 1>
    - <visual feature 2>
```

This enables the LLM to output useful visual features that characterize the class in phrases. Actual LLM outputs can be found in the supplemental material.

**Filtering Visually-Non-matched Textual Features.** With visual features produced by the LLM, we utilize pre-trained CLIP encoders to filter out irrelevant features that do not align with the given image. To achieve this, each visual feature is encapsulated within a phrase structured as follows:

The photo of {cls} which has {viz\_feat}.

The prompted sentences are then fed into the text encoder, while the target image is fed into the image encoder at the same time. The output embeddings are used to compute the logit scores that indicate the similarities between the image and each sentence. Among the visual features generated in the first stage, only the ones with scores higher than the predefined threshold are selected as final features to be incorporated into the descriptions.

**Complete Sentence Generation.** In the final stage, we employ LLM once again to obtain natural and realistic sentences. We first construct a simple sentence with final features obtained from previous stage. Assuming two visual features `viz_feat.1` and `viz_feat.2`, the sentence is formed as follows:

It has {viz\_feat\_1} and {viz\_feat\_2}.

Next, to gather a more diverse and varied set of expressions, we prompt the LLM with another prompt:

Refine this to a standard English:  
<simple sentence>

Finally, the resulting sentence becomes the final description for the image. Examples of generated descriptions are shown in Fig. 4.

## 5 Experiments

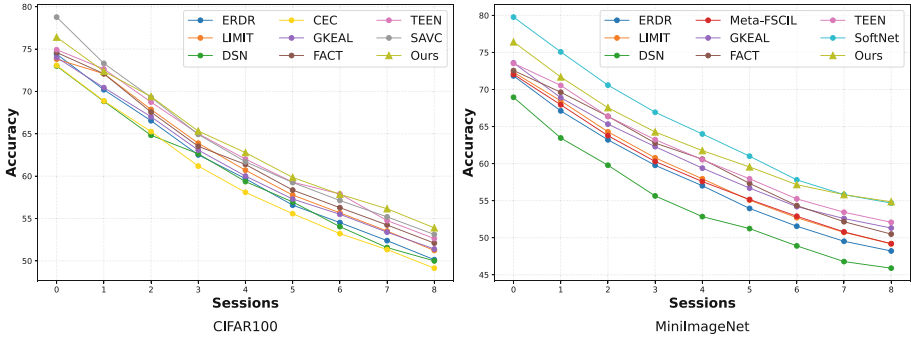
### 5.1 Datasets and Settings

**Implementation Details.** We employ ResNet-18 as our visual encoder, since ResNet [9] is commonly used as a backbone network for addressing the FSCIL problem. Consistent with prior setups [21, 26, 34, 37], we initiate the training of our visual encoder from scratch for CIFAR100 [14] and miniImageNet [28]. Meanwhile, for the CUB200 [29] dataset, we utilize a pre-trained ResNet on ImageNet as initialization. Additionally, we leverage a frozen pre-trained CLIP [22] model as the textual encoder, which produces 512-dimensional textual latent representations. To align the representation spaces of both modalities and restore their representation power, we employ projection layers. Following previous works [21, 34], these projection layers consist of two-layer MLP blocks, producing 2048-dimensional representation for each. Our model is trained for 100 epochs on a single NVIDIA A100 GPU using an SGD optimizer with a learning rate of 0.01 (0.001 on CUB200), a momentum of 0.9, and a weight decay rate of  $5e-4$ .

**Table 1.** We compare our model with the state-of-the-art methods in terms of performance dropping rate (PD, in %) and average accuracy (AA, in %) on CUB200 benchmark. Our method outperforms existing works with significant performance gaps. Note that “S0” indicates the accuracy in the base session. Methods are sorted in descending order of PD scores.

Methods	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	AA(†)	PD(↓)
TOPIC [26]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	42.40
Cheraghian <i>et al.</i> [4]	68.23	60.45	55.70	50.45	45.72	42.90	40.89	38.77	36.51	34.87	32.96	46.13	35.27
SPPR [43]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	31.35
CEC [37]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	23.57
ERDR [15]	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	23.51
Meta-FSCIL [6]	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	61.93	23.26
DSN [33]	76.06	72.18	69.57	66.68	64.42	62.12	60.16	58.94	56.99	55.10	54.21	63.31	21.85
CaBD [39]	79.12	74.99	70.87	67.30	65.89	63.45	61.40	60.11	58.61	58.23	57.48	65.22	21.64
SoftNet [12]	78.07	74.58	71.37	67.54	65.37	62.6	61.07	59.37	57.53	57.21	56.75	64.68	21.32
NC-FSCIL [34]	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	21.01
WaRP [13]	77.74	74.15	70.82	66.9	65.01	62.64	61.40	59.86	57.95	57.77	57.01	64.66	20.73
GKEAL [44]	78.88	75.62	72.32	68.62	67.23	64.26	62.98	61.89	60.20	59.21	58.67	66.35	20.21
CLOM [45]	79.57	76.07	72.94	69.82	67.80	65.56	63.94	62.59	60.62	60.34	59.58	67.17	19.99
FSSL [18]	75.63	71.81	68.16	64.32	62.61	60.10	58.82	58.70	56.45	56.41	55.82	62.62	19.81
CSR [40]	74.69	71.29	67.82	64.41	62.41	60.20	59.06	58.16	56.37	55.99	55.09	62.32	19.60
SAVC [25]	81.85	77.92	74.95	70.21	69.96	67.02	66.16	65.30	63.84	63.15	62.50	<b>69.35</b>	19.35
FACT [41]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	18.96
MgSvF [38]	72.29	70.53	67.00	64.92	62.67	61.89	59.63	59.15	57.73	55.92	54.33	62.37	17.96
TEEN [31]	77.26	76.13	72.81	68.16	67.77	64.40	63.25	62.29	61.19	60.32	59.31	66.63	17.95
LIMIT [42]	76.32	74.18	72.68	69.19	68.79	65.64	63.57	62.69	61.47	60.44	58.45	66.67	17.87
ALICE [21]	77.40	72.70	70.60	67.20	65.90	63.40	62.90	61.90	60.50	60.60	60.10	65.75	17.30
Ours	78.98	76.49	73.91	70.47	68.89	66.76	65.86	65.12	63.45	63.58	62.77	68.75	<b>16.21</b>

**Datasets.** To evaluate the effectiveness of text-driven prototype learning, we use three widely-used benchmark datasets: CUB200 [29], CIFAR100 [14], and miniImageNet [28]. CUB200 dataset provides over 11k images for 200 classes of North American bird species, and has a collection of descriptions with ten descriptive sentences per image [23]. These sentences convey a detailed class-specific and class-discriminative description about why the bird is classified in a particular class, *e.g.*, “this bird has a black crown, a white eye and a large black bill.” Following existing FSCIL works, we set 11 consecutive sessions, including the base session (called session 0). Half of the whole 200 classes are consumed during the base session. In the following incremental sessions (*i.e.*, from session 1 to session 10), the rest of the 100 classes are equally divided into ten disjoint subsets (*i.e.*, ten classes for each session) and five randomly chosen samples are provided for each class (*i.e.*, 10-way 5-shot). Unlike CUB200 dataset, CIFAR100 and miniImageNet datasets do not have such descriptions with their public datasets. Thus, as mentioned earlier, we create new text datasets that contain a single descriptive sentence per train image. In each dataset, 100 classes are divided such that 60 serve as base classes and the remaining 40 are designated as



**Fig. 5.** Performance comparison with state-of-the-art approaches on two widely-used benchmarks: CIFAR100, and miniImageNet. We provide detailed performance table in the supplemental material.

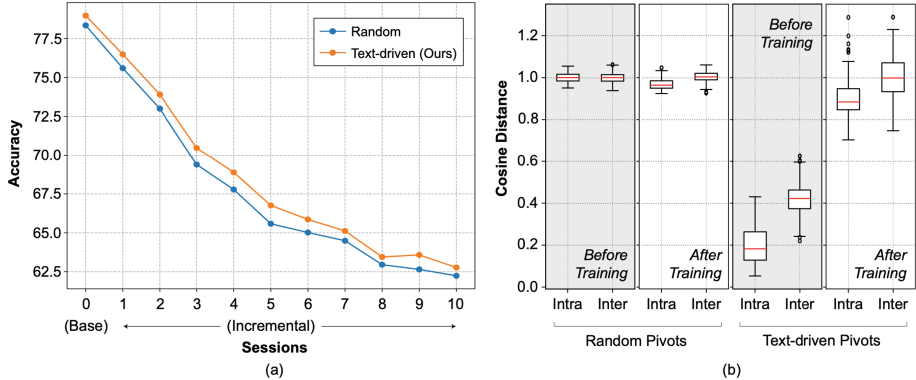
new classes for use in subsequent incremental sessions. Following existing FSCIL works, we use an 8-step 5-way 5-shot setup.

## 5.2 Text-Driven Prototype Analysis

**Comparison with State-of-the-Art Approaches.** To evaluate our model against existing methods, we employ two widely-used metrics: average accuracy (AA) and performance dropping rate (PD). The former calculates the mean accuracy across all sessions, whereas the latter quantifies the total drops in accuracy from the final session compared to the base session. Ideally, a model would exhibit both high average accuracy and a low performance dropping rate. As shown in Table 1, our method surpasses existing state-of-the-art approach on CUB200 with the best PD score of 16.21%, indicating minimal knowledge forgetting. Moreover, Fig. 5 shows that our model’s performance is comparable to that of state-of-the-art methods on CIFAR100 and miniImageNet benchmarks as well. A detailed performance table for Fig. 5 is provided in the supplemental material.

**Effect of Text-Driven Prototype Learning.** We further analyze the effect of using text-driven prototypes against random-initialized prototypes. For random-initialized prototypes, we use arbitrary vectors that are uniformly scattered in a circular distribution. As shown in Fig. 6 (a), we observe that text-driven prototype initialization clearly outperforms random initialization with a large gap on CUB200 dataset: an average accuracy improves 0.84%.

**t-SNE Analysis.** In Fig. 1 (rightmost column), we visualize two different embedding spaces (*i.e.*, randomly initialized and text-driven) using t-SNE [17] on the CUB200 dataset. Compared to the randomly initialized model (top), our model exhibits fewer false positives (yellow dots in the cluster of B, C, and D) and more distinguishable clusters for each class, when visually-similar classes



**Fig. 6.** (a) Performance comparison using different prototype initialization methods. Our text-driven prototype outperforms the baseline model (*i.e.*, random initialization). (b) Box-plots of cosine distances between pivots that are initialized randomly (left) or text-driven pivots (right). We provide box plots for before- and after-base sessions. Note that we use the CUB200 [29] dataset for these experiments.

(*i.e.*, B, C, and D) of the base class (*i.e.*, A) are added during the incremental session. This may confirm the effectiveness of leveraging text-driven prototypes in the FSCIL task for learning better embedding space and maintaining semantic relations between classes.

**Semantic-Distilled Prototypes.** Figure 6 (b) shows side-by-side box plots for randomly-initialized prototypes (left) and text-driven prototypes (right) in the CUB200 dataset. Each set includes two box plots representing cosine distances between semantically similar classes (denoted as ‘Intra’) and semantically different classes (denoted as ‘Inter’), with distances calculated by subtracting the cosine similarity from 1. Note that semantically similar classes are defined based on the class names. Specifically, CUB200 class names consist of two parts, with the latter part indicating their higher-level categories. We categorize classes with identical higher-level categories as ‘semantically similar classes (Intra classes)’ and all other classes as ‘semantically unrelated classes (Inter classes)’. Remarkably, randomly-initialized prototypes exhibit cosine distances closer to 1 (orthogonal), indicating all prototypes are irrelevant. In contrast, text-driven prototypes show a broader range of cosine distances, which tend to be closer to semantically similar classes compared to semantically unrelated classes. As our experiments and previous studies (*e.g.*, [24]) suggest, this semantic distillation can help the improvement of generalization capabilities.

## 6 Discussions

In this work, we explore the benefits of leveraging textual modality to improve FSCIL tasks and observe substantial performance improvements. However, our method inherently depends on the quality of text inputs; specifically, the texts

should be class-discriminative and visually descriptive, which may not always be feasible in real-world settings. Thus, we create new textual datasets, CIFAR100-Text and miniImageNet-Text, and utilizing LLMs similar to our dataset collection approach may help mitigate this limitation. Furthermore, while we follow the standard setting for FSCIL tasks, it may not be sufficiently flexible for deployment in real-world applications. Addressing this challenge could direct our future work, exploring effective ways to leverage textual semantics in more complex, diverse, and practical scenarios.

## 7 Conclusion

In this paper, we propose a text-based prototype learning approach for few-shot class-incremental learning (FSCIL). We leverage textual semantics by utilizing angular margin loss so that text-driven prototypes retain semantic relations between all old and new classes, encouraging intra-class compactness and inter-class discrepancies in the embedding space. Our experiments with three popular FSCIL benchmarks (CUB200, CIFAR100, and miniImageNet) confirm the effectiveness of using text modality in generating prototypes, where our model generally matches or outperforms the current state-of-the-art approaches. Moreover, we newly collect visually descriptive and class-discriminate descriptions using a GPT-3-based large language model built upon two widely-used FSCIL benchmarks: CIFAR100-Text and miniImageNet-Text.

**Acknowledgments.** This work was supported by Samsung Electronics. Also, this work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A13044830, 15%) and supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2022-II220043, Adaptive Personality for Intelligent Agents, 15%, IITP-2024-RS-2024-00397085, Leading Generative AI Human Resources Development, 15%).

## References

1. Brown, T., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
2. Chen, K., Lee, C.G.: Incremental few-shot learning via vector quantization in deep embedded space. In: *International Conference on Learning Representations* (2021)
3. Chen, W., Si, C., Zhang, Z., Wang, L., Wang, Z., Tan, T.: Semantic prompt for few-shot image recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23581–23591 (2023)
4. Cheraghian, A., Rahman, S., Fang, P., Roy, S.K., Petersson, L., Harandi, M.: Semantic-aware knowledge distillation for few-shot class-incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2534–2543 (2021)

5. Cheraghian, A., et al.: Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8661–8670 (2021)
6. Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., Tang, J.: Metafscl: a meta-learning approach for few-shot class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14166–14175 (2022)
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition, pp. 4690–4699 (2019)
8. Dong, S., Hong, X., Tao, X., Chang, X., Wei, X., Gong, Y.: Few-shot class-incremental learning via relation knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 1255–1263 (2021)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
10. Hersche, M., Karunaratne, G., Cherubini, G., Benini, L., Sebastian, A., Rahimi, A.: Constrained few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9057–9067 (2022)
11. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning, pp. 4904–4916. PMLR (2021)
12. Kang, H., Yoon, J., Madjid, S.R.H., Hwang, S.J., Yoo, C.D.: On the soft-subnetwork for few-shot class incremental learning (2023)
13. Kim, D.Y., Han, D.J., Seo, J., Moon, J.: Warping the space: Weight space rotation for class-incremental few-shot learning. In: The Eleventh International Conference on Learning Representations (2023). <https://openreview.net/forum?id=kPLzOfPFA2l>
14. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009). <https://api.semanticscholar.org/CorpusID:18268744>
15. Liu, H., Gu, L., Chi, Z., Wang, Y., Yu, Y., Chen, J., Tang, J.: Few-shot class-incremental learning via entropy-regularized data-free replay. arXiv preprint [arXiv:2207.11213](https://arxiv.org/abs/2207.11213) (2022)
16. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 212–220 (2017)
17. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9** (2008)
18. Mazumder, P., Singh, P., Rai, P.: Few-shot lifelong learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2337–2345 (2021)
19. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology Of Learning And Motivation*, vol. 24, pp. 109–165. Elsevier (1989)
20. Menon, S., Vondrick, C.: Visual classification via description from large language models. arXiv preprint [arXiv:2210.07183](https://arxiv.org/abs/2210.07183) (2022)



21. Peng, C., Zhao, K., Wang, T., Li, M., Lovell, B.C.: Few-shot class-incremental learning from an open-set perspective. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pp. 382–397. Springer (2022). [https://doi.org/10.1007/978-3-031-19806-9\\_22](https://doi.org/10.1007/978-3-031-19806-9_22)
22. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR (2021)
23. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58 (2016)
24. Roth, K., Vinyals, O., Akata, Z.: Integrating language guidance into vision-based deep metric learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16177–16189 (2022)
25. Song, Z., Zhao, Y., Shi, Y., Peng, P., Yuan, L., Tian, Y.: Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning (2023)
26. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192 (2020)
27. Tian, S., Li, L., Li, W., Ran, H., Ning, X., Tiwari, P.: A survey on few-shot class-incremental learning. *arXiv preprint arXiv:2304.08130* (2023)
28. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. *Adv. Neural Inform. Process. Syst.* **29** (2016)
29. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
30. Wang, H., et al.: Cosface: large margin cosine loss for deep face recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274 (2018)
31. Wang, Q.W., Zhou, D.W., Zhang, Y.K., Zhan, D.C., Ye, H.J.: Few-shot class-incremental learning via training-free prototype calibration (2023)
32. Xing, C., Rostamzadeh, N., Oreshkin, B., O Pinheiro, P.O.: Adaptive cross-modal few-shot learning. *Adv. Neural Inform. Process. Syst.* **32** (2019)
33. Yang, B., et al.: Dynamic support network for few-shot class incremental learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
34. Yang, Y., Yuan, H., Li, X., Lin, Z., Torr, P., Tao, D.: Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In: *The Eleventh International Conference on Learning Representations* (2023)
35. Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., Yatskar, M.: Language in a bottle: language model guided concept bottlenecks for interpretable image classification. *arXiv preprint arXiv:2211.11158* (2022)
36. Yao, G., Zhu, J., Zhou, W., Li, J.: Few-shot class-incremental learning based on representation enhancement. *J. Electron. Imaging* **31**(4), 043027 (2022)
37. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., Xu, Y.: Few-shot incremental learning with continually evolved classifiers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12455–12464 (2021)
38. Zhao, H., Fu, Y., Kang, M., Tian, Q., Wu, F., Li, X.: Mgsvf: multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021). <https://doi.org/10.1109/TPAMI.2021.3133897>

39. Zhao, L., et al.: Few-shot class-incremental learning via class-aware bilateral distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11838–11847 (June 2023)
40. Zheng, G., Zhang, A.: Few-shot class-incremental learning with meta-learned class structures. In: 2021 International Conference on Data Mining Workshops (ICDMW), pp. 421–430. IEEE (2021)
41. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9046–9056 (2022)
42. Zhou, D.W., Ye, H.J., Ma, L., Xie, D., Pu, S., Zhan, D.C.: Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022)
43. Zhu, K., Cao, Y., Zhai, W., Cheng, J., Zha, Z.J.: Self-promoted prototype refinement for few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6801–6810 (2021)
44. Zhuang, H., Weng, Z., He, R., Lin, Z., Zeng, Z.: Gkeal: gaussian kernel embedded analytic learning for few-shot class incremental task. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7746–7755 (2023)
45. Zou, Y., Zhang, S., Li, Y., Li, R.: Margin-based few-shot class-incremental learning with class-level overfitting mitigation. arXiv preprint [arXiv:2210.04524](https://arxiv.org/abs/2210.04524) (2022)



# Dual Supervised Contrastive Learning Based on Perturbation Uncertainty for Online Class Incremental Learning

Shibin Su<sup>1</sup>, Zhaojie Chen<sup>1</sup>, Guoqiang Liang<sup>2,1,\*</sup>, Shizhou Zhang<sup>1</sup>,  
and Yanning Zhang<sup>1</sup>

<sup>1</sup> School of Computer Science, Northwestern Polytechnical University, Xi'an, China  
gqliang@nwpu.edu.cn

<sup>2</sup> Research & Development Institute of Northwestern Polytechnical  
University in Shenzhen, Shenzhen 518063, China<sup>2,1,\*</sup>

**Abstract.** To keep learning knowledge from a data stream with changing distribution, continual learning has attracted lots of interests recently. Among its various settings, online class-incremental learning (OCIL) is more realistic and challenging since the data can be used only once. Currently, by employing a buffer to store a few old samples, replay-based methods have obtained huge success and dominated this area. Due to the single pass property of OCIL, how to retrieve high-valued samples from memory is very important. In most of the current works, the logits from the last fully connected layer are used to estimate the value of samples. However, the imbalance between the number of samples for old and new classes leads to a severe bias of the FC layer, which results in an inaccurate estimation. Moreover, this bias also brings about abrupt feature change. To address this problem, we propose a dual supervised contrastive learning method based on perturbation uncertainty. Specifically, we retrieve samples that have not been learned adequately based on perturbation uncertainty. Retraining such samples helps the model to learn robust features. Then, we combine two types of supervised contrastive loss to replace the cross-entropy loss, which further enhances the feature robustness and alleviates abrupt feature changes. Extensive experiments on three popular datasets demonstrate that our method surpasses several recently published works.

**Keywords:** Online class-incremental learning · Perturbation uncertainty retrieval · Supervised contrastive learning

## 1 Introduction

When data are drawn from an independently and identically distribution (i.i.d), Deep Neural Networks (DNNs) have demonstrated excellent performance in numerous tasks. However, in many practical scenarios, the distribution of data

---

S. Su and Z. Chen—Authors contributed equally.

stream are changing continuously. If we directly adopt current models, catastrophic forgetting [31] will occur with very high probability. In detail, the performance of old tasks reduces heavily after the model is trained on new tasks. Therefore, continual learning, which can keep learning and accumulating knowledge from a never-ending data stream, has attracted a lot of interest [30, 35].

Currently, continual learning scenarios can be roughly classified into three types: task incremental learning, domain incremental learning, and class incremental learning [35]. In class incremental learning [1, 11], the model should predict its class given an input from an arbitrarily trained class. Since task identification is not available and new classes appear continuously, it is more realistic. In this work, we further focus on a more challenging setting-online class incremental learning (OCIL) [10, 11, 29, 33]. Compared with the offline class incremental learning where the data stream of a task can be used repeatedly, the online data stream can be used only once.

Replay is a commonly used strategy to alleviate catastrophic forgetting and has achieved great success in continual learning [2, 3, 6, 11]. In detail, a small number of data from old tasks are stored in a memory buffer. When a new task comes, some samples are retrieved from the memory and combined with new data to update the model. Thus, the memory retrieval strategy plays an important role. Currently, many replay-based methods [2, 33] employ logits from the last FC layer to measure the sample value. However, there exists a severe class imbalance problem due to the limit of memory size [1, 25]. Specifically, the data for new tasks is much more than that of old tasks. This class imbalance will lead to the task-recency bias of the FC layer, which affects the accuracy of measurement. Moreover, this imbalance also causes an abrupt change of feature representation during model learning, leading to catastrophic forgetting [28, 30]. To alleviate this, many works focus on how to learn robust feature representation [8] or alleviate task-recency bias [25, 29]. The former aims to prevent abrupt feature change of old classes when the model adapts to the feature space of new tasks. The latter prevents the weights for old classes in the FC layer from being penalized during the gradient descent. Although they obtain huge progress, the class imbalance still exists for the combination of FC and soft-max.

To address the above issues, we propose a dual supervised contrastive learning method based on perturbation uncertainty. Considering the shortcoming of the estimation of the sample value based on logits, we design a perturbation uncertainty-based retrieval strategy, which aims to retrieve memory samples that have not been learned adequately. In detail, we regard the similarity of features of an original sample and its augmented one as the score of its uncertainty. A lower similarity score indicates that the model is more uncertain about this sample. In other words, retraining on these samples will enhance the robustness of the feature. Furthermore, to mitigate abrupt feature change and task-recency bias caused by traditional cross-entropy loss, we replace it with the combination of two types of supervised contrastive learning loss. One is based on sample-to-sample relation, which encourages the feature of the same class's samples to be clustered closely. However, it is easy to hurt the ability of model to learn

new classes. So we also employ the proxy-to-sample based supervised contrastive learning loss. Combining them can offset their disadvantages and obtain a better feature representation. Like previous paper [41], we use the proxy to denote the representative of a subset for a single class, which is robust to noisy samples or outliers. Currently, there are two different ways to calculate the proxy of each class, the mean feature or normalized FC weight of the last layer. Since the data of each class can be used only once in OCIL, it is very tough to learn a robust feature representation. Furthermore, using normalized FC weight does not increase computational cost compared to using mean feature representation. Therefore, we employ the FC weights as our proxy. In the proxy-to-sample based supervised contrastive learning loss, the embeddings and normalized FC weights are used to compute the loss. Normalized FC weights enable the proxy-to-sample based supervised contrastive learning loss to effectively mitigate the task-recency bias associated with traditional cross entropy loss.

To evaluate the proposed method, we have conducted extensive experiments on three popular datasets: Split CIFAR10 [22], Split CIFAR100 [22] and split Mini-ImageNet [36]. The results show the effectiveness of the model.

Our main contributions can be summarized as follows.

1) We propose a simple but effective memory retrieval strategy to seek samples that have not been learned adequately, which enhances the robustness of the feature.

2) We replace the cross-entropy loss with the combination of two types of supervised contrastive learning loss, which alleviates task-recency bias.

3) We conduct extensive experiments on three popular continual learning benchmarks, where the proposed model outperforms several recently published replay-based methods.

## 2 Related Work

### 2.1 Continual Learning

Recent methods for continual learning can be mainly divided into three classes: regularization, dynamic architecture, and memory replay. In regularization-based methods, various regularization terms are designed to penalize updating of important parameters previously learned [21, 26]. Among them, knowledge distillation (KD) is an important way which has been applied to preserve probability, logit, feature, relation, etc. [23, 24, 34]. In dynamic architecture-based methods, the model is dynamically expanded [14] or specific parameters are allocated for the new tasks [4, 32]. By utilizing a memory buffer, replay-based methods can store and replay a small number of previous samples to better preserve old knowledge. Recently, as prompt tuning [19] has become popular, some works have introduced it into continual learning [39, 40].

Since our method is based on replay, we mainly review replay-based methods. Refer to [30, 35] for details of other approaches. In early replay-based methods, such as GEM [27] and A-GEM [10], the memory buffer served as a constraint in

gradient updates to ensure that the loss for buffer samples did not increase. However, these methods, which did not train directly on the memory buffer, underperformed when compared to experience replay methods. In experience replay (ER) [11], the raw data samples were stored in memory and trained directly with the samples of the current task together. Due to its effectiveness and simplicity, numerous variants were developed [5–8]. For instance, DER [6] and X-DER [5] stored the logits with the previous samples and used logits-matching during the optimization procedure. ER-ACE [8] employed an asymmetric cross-entropy loss to improve the quality of the representations. Moreover, SCR [29], Co2L [9], DVC [16] and ER-AML [8] incorporated the contrastive learning to learn robust feature. [15] compared the effectiveness of three contrastive learning methods in offline and online continual learning. [13] proposed a fast remembering method by combining fine-tuning based on supervised contrastive learning with a small memory. To correct the distribution shift online, [38] integrated a Continual Bias Adaptor (CBA) module into replay-based methods. On the other hand, since each sample can only be trained once in OCIL, memory sample selection is also a crucial consideration. Instead of random retrieval, MIR [2] retrieved samples based on the loss change of the current mini-batch while ASER [33] utilized the adversarial Shapley value to make model update more effective. Moreover, although most replay-based models store samples randomly, some studies such as GSS [3], OCS [42] try to select the most representative data for storing.

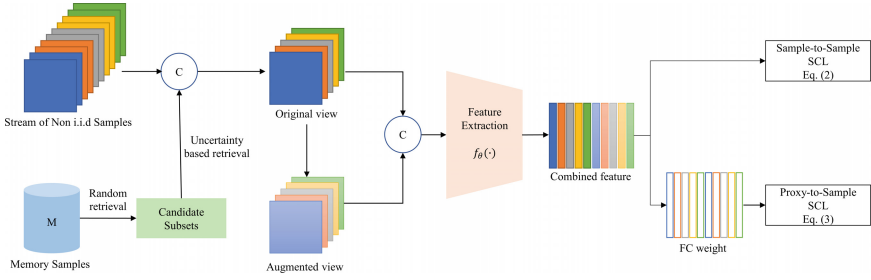
## 2.2 Contrastive Learning

Contrastive learning can be traced back to [17], which was originally developed for dimensionality reduction. By comparing the similarities of different images pairs, contrastive learning aims to learn excellent feature representations. In recent years, contrastive learning has achieved significant progress with the help of deep learning. By comparing a large number of positive and negative sample pairs, deep contrastive learning models can extract better feature representations. SimCLR [12] and MoCo [18] are two typical methods, both of which are self-supervised. By designing a supervised contrastive learning loss, Khosla et al. [20] extended contrastive learning to the supervised learning area. Through leveraging label information, it largely improves the feature representation. Yao et al. [41] replaced the traditional sample-to-sample relations with the proxy-to-sample relation model, which helps to align some difficult positive sample pairs with larger difference.

Different from the above works, we integrate two types of supervised contrastive loss for continual learning, which enhances the robustness of the features and alleviates abrupt feature changes.

## 3 Method

In this section, we will detail the main modules of the proposed method after a brief introduction of the definition of the OCIL problem.



**Fig. 1.** Overview of the proposed method. Firstly, the uncertainty based retrieval is used to select some meaningful samples from memory, which are then concatenated with samples from data stream. Then, these samples and their augmented view are fed to the feature extractor  $f_{\theta}(\cdot)$ . Finally, we employ two types of supervised contrastive loss (Eq. 2 and Eq. 3) to optimize the whole parameters.

### 3.1 Problem Definition

In OCIL, a model needs to continually learn new classes from an online non-i.i.d data stream  $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T\}$ , where  $T$  denotes the total number of tasks. Each task  $D_t = \{(x_{t,n}, y_{t,n})\}_{n=1}^{N_t} \sim \mathcal{X}_t \times \mathcal{Y}_t$  comprises independently and identically distributed data  $\mathcal{X}_t$  paired with associated label  $\mathcal{Y}_t$ . Note that each task contains non-overlapping classes, i.e.  $Y_i \cap Y_j = \emptyset$  for  $\forall i, j \in \{1, \dots, T\}, i \neq j$ . Furthermore, several pairs of data and labels can be packaged into mini-batches for training, which can be denoted as  $D_t = \{B_{t,1}, B_{t,2}, \dots, B_{t,k}\}$ , where each batch contains  $b$  samples. Note that each batch can be passed to the model only once. Once data have been processed, they cannot be revisited unless they are stored in memory buffer  $\mathcal{M}$ . The model consists of a feature extraction module  $f_{\theta}(\cdot)$  and a classifier  $g(\cdot)$ . This work focuses exclusively on a single-head classifier.

### 3.2 Method Overview

Figure 1 illustrates the overview of the proposed method for model training. To select samples with higher values, we first create a candidate subset from the memory through random retrieval. Then, the proposed perturbation uncertainty based retrieval strategy is applied to seek more meaningful samples, which are then combined with new task samples. These samples and their augmented view are fed to the network to produce logits. To learn robust features and improve the model stability, we combine two types of supervised contrastive loss during model learning. One works on the sample-to-sample relations while the other models the proxy-to-sample relations. After each batch, the reservoir sampling algorithm [37] is used to randomly select some samples from current batch to update the memory. During testing, each sample is fed to the feature extraction network and the classifier to generate the final probability.

In the following, we focus on the two main modules: the perturbation uncertainty based retrieval and supervised contrastive learning loss.

### 3.3 Perturbation Uncertainty Based Memory Retrieval

Although random memory retrieval has been widely applied in continual learning, how to search for valuable samples still plays an important role. When the model learns a new task, the sample feature of the old tasks will drift. And the more severe the drift, the more likely the false prediction. Meanwhile, some samples' feature may be unaffected and retraining on them brings little contribution. Therefore, we want to select the samples whose features are more likely to move. However, the future model is unknown. Although virtual update [2] seems a feasible method, it requires high computational cost. Instead, we propose a perturbation uncertainty based memory retrieval strategy, which measures the feature similarity between an original sample and its perturbed view to retrieve meaningful samples.

Specifically, we first randomly retrieve a candidate subset  $B_{cand} = (x_i, y_i)_{i=1}^{sz}$  from memory buffer, where  $sz$  is the total number of retrieved samples. Compared with measuring all samples, this subset can reduce computation largely. Subsequently, for each sample in this candidate subset, we perform data augmentation  $Aug(\cdot)$  to generate an augmented view, which leads to an augmented subset  $\hat{B}_{cand} = (Aug(x_i), y_i)_{i=1}^{sz}$ . Then, for each sample, we calculate the feature similarity between the original data and its augmented view as its uncertainty measurement. Formally, for the  $i$ -th sample  $x_i$ , the uncertainty score can be computed as

$$\text{score}_i = 1.0 - \frac{f_\theta(x_i) \cdot f_\theta(Aug(x_i))}{|f_\theta(x_i)| \cdot |f_\theta(Aug(x_i))|} \quad (1)$$

where  $|\cdot|$  denotes the L2 normalization. According to this equation, a bigger score indicates that the features representation before and after augmentation are more different. In other words, the feature of samples with bigger uncertainty scores is more likely to change after model update.

If we directly select the  $k$  samples with the highest scores, the diversity of the resulting samples is limited. Moreover, overemphasizing these challenging samples will damage the model's ability to learn new tasks. To avoid this, we first rank all the candidate samples according to the score and then select  $k$  samples, with an interval of  $sz/k$ , from the samples with the biggest score. This operation can not only increase the diversity of selected samples but also reduce the difficulty of model learning.

### 3.4 Supervised Contrastive Learning

Due to the class imbalance, the traditional combination of the FC layer and cross-entropy loss will lead to task recency bias and abrupt feature change. On the other hand, supervised contrastive learning (SCL) has made great progress in learning a robust feature [41]. Specifically, it can cluster the features of the same class and separate the features of different classes. Inspired by this, we propose to replace the widely used cross-entropy loss with the supervised contrastive loss, which can make the feature more robust.



The popular SCL framework [20] consists of three modules, i.e. data augmentation  $Aug(\cdot)$ , feature extraction network  $f_\theta(\cdot)$ , and projection layers  $Proj(\cdot)$ . For a sample  $x$ , the augmentation produces an augmented view  $\tilde{x} = Aug(x)$  while the feature extraction network maps an input to a feature representation  $z = f_\theta(x)$ . However, in our initial experiments, the projection layers have little influence on the OCIL performance [29]. Thus, we drop it to reduce the computation complexity. In this way, for a batch  $B$ , we first perform data augmentation to generate its augmented one  $\tilde{B}$ , which is then concatenated with the original batch, i.e.  $B_I = B \cup \tilde{B}$ . Note that the original batch  $B$  is the concatenation of a current data batch and a retrieved memory batch. Finally, the joint batch  $B_I$  is fed to the feature extractor  $f_\theta(\cdot)$  to obtain their feature representation  $Z_I$ .

To optimize the model’s parameters, we combine two types of supervised contrastive learning loss. The first one is based on sample-to-sample relations, which can be formulated as

$$L_{SS}(Z_I) = \sum_{z_i \in Z_I} \frac{-1}{|P(i)|} \sum_{z_p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{z_j \in Z_I, j \neq i} \exp(z_i \cdot z_j / \tau)} \quad (2)$$

where  $\tau$  is the temperature scale,  $z_i$  is the feature of  $i$ -th sample in the joint batch  $B_I$ ,  $P(i)$  denotes the feature set of all other samples in  $B_I$  which have the same class label with  $x_i$ . According to Eq. (2), we aim to cluster the features of the same class and separate the features of different classes.

Although the above loss captures the rich sample-to-sample relations, it is hard to converge because of the complex sample-to-sample relationship and the single pass in OCIL. Moreover, when the distribution gap is larger, it tends to hinder the generalization of the model [41]. Considering this, we also employ the new proposed supervised contrastive loss based on proxy-to-sample relations [41], where the proxy is regarded as the representative of a sub-dataset. Formally, this PSCL loss can be written as

$$L_{PS}(Z_I) = \sum_{z_i \in Z_I} -\log \frac{\exp(z_i \cdot w_i / \tau)}{\sum_{j \in C_B} \exp(z_i \cdot w_j / \tau)} \quad (3)$$

where  $C_B$  is the classes indices in the joint batch,  $w_j$  denotes the proxy of  $j$ -th class. Note that the  $w_i$  in the numerator is the proxy corresponding to the class label of sample  $x_i$ . In our implementation, we directly adopt the weight vector corresponding to a class in the FC classifier as its proxy. According to this equation, this proxy can ensure stable and fast convergence.

To guarantee the diversity and stability of the model, we combine the two supervised contrastive losses as following

$$L_{\text{total}} = L_{PS} + \beta L_{SS} \quad (4)$$

where  $\beta$  is a hyper-parameter to balance the two losses.

The whole procedure of the proposed method is shown in Algorithm 1. During training, we optimize the parameters of feature extractor and the weights in the FC classifier. For inference, we first extract the feature and then use the FC classifier and soft-max to produce final prediction.

**Algorithm 1.** Dual Contrastive Learning Based on Perturbation Uncertainty

---

**Input:** Memory Buffer Size  $M$ , Data Augmentation  $Aug(\cdot)$ , Feature Extraction  $f_\theta(\cdot)$   
Parameterized by  $\theta$ , FC classifier  $g_w(\cdot)$ , Learning Rate  $\lambda$

**Initialization:** Memory Buffer  $\mathcal{M} \leftarrow \{\} * M$ , Number of Observed Sample  $n \leftarrow 0$

- 1: **for**  $T \in \{1, \dots, N\}$  **do**
- 2:   // Training Phase
- 3:   **for**  $B_i \sim \mathcal{D}_T$  **do**
- 4:      $B_{cand} \leftarrow \text{RandomRetrieval}(\mathcal{M})$
- 5:      $Score_i \leftarrow 1.0 - \frac{f_\theta(x_i) \cdot f_\theta(Aug(x_i))}{|f_\theta(x_i)| \cdot |f_\theta(Aug(x_i))|}, x_i \in B_{cand}$
- 6:     Rank the candidate sample set  $B_{cand}$  according to  $Score$
- 7:     Construct set  $B_{\mathcal{M}}$  by selecting samples from  $B_{cand}$  by every  $\lfloor \frac{sz}{k} \rfloor$  interval
- 8:      $B_I \leftarrow (B_i \cup B_{\mathcal{M}}) \cup Aug(B_i \cup B_{\mathcal{M}})$
- 9:      $Z_I \leftarrow f_\theta(B_I)$
- 10:      $L_{SS}(Z_I) = \sum_{z_i \in Z_I} \frac{-1}{|P(i)|} \sum_{z_p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{z_j \in Z_I, j \neq i} \exp(z_i \cdot z_j / \tau)}$
- 11:      $L_{PS}(Z_I) = \sum_{z_i \in Z_I} -\log \frac{\exp(z_i \cdot w_i / \tau)}{\sum_{j \in C_B} \exp(z_i \cdot w_j / \tau)}$
- 12:      $L \leftarrow L_{PS} + \beta L_{SS}$
- 13:      $\theta, w \leftarrow \text{SGD}(L, \theta, w, \lambda)$
- 14:      $\mathcal{M} \leftarrow \text{ReservoirUpdate}(\mathcal{M}, B_i, M, n)$
- 15:      $n \leftarrow n + |B_i|$
- 16:   **end for**
- 17: **end for**
- 18: // Inference Phase
- 19: **for**  $i \in \{1, \dots, N_{test}\}$  **do**
- 20:    $z_i = f_\theta(x_i)$
- 21:    $y_i^* \leftarrow \arg \max_{c \in C_{1:T}} \frac{\exp(w_c \cdot z_i / \tau)}{\sum_{j \in C_{1:T}} \exp(w_j \cdot z_i / \tau)}$
- 22: **end for**

---

## 4 Experiment

In this section, we first introduce the benchmark datasets, metrics, and implementation details. Then, we will report and analyze the experimental results.

### 4.1 Experiment Setup

**Datasets.** Following [2, 16, 29, 33], we adopt the three commonly used datasets for continual learning: Split CIFAR10 [22], Split CIFAR100 [22] and split Mini-ImageNet [36]. In split-CIFAR10, the dataset is split into different 5 tasks and each task contains 2 classes. For split CIFAR100 and split Mini-ImageNet, we generate 10 disjoint tasks, each of which contains 10 classes.

**Metrics.** We primarily employ the average accuracy (AA) and average forgetting (AF) as the metrics, which are defined as

$$AA(A_i) = \frac{1}{T} \sum_{j=1}^T a_{i,j} \quad (5)$$

$$AF(F_i) = \frac{1}{i-1} \sum_{j=1}^{T-1} f_{i,j} \text{ s.t. } f_{i,j} = \max_{l \in \{1, \dots, i-1\}} a_{l,j} - a_{i,j} \quad (6)$$

where  $a_{i,j}$  and  $f_{i,j}$  denotes the accuracy and forgetting rate of task  $j$  after training on the task  $i$  respectively. When  $i = T$ , the  $A_T$  and  $F_T$  represent the final average accuracy (FAA) and final average forgetting (FAF) respectively. Obviously, the larger FAA and the smaller FAF, the better performance.

In addition, we use the average accuracy of new class (AAN) to evaluate the plasticity of a model following [44].

$$AAN(AN_i) = \frac{1}{T} \sum_{i=1}^T a_{i,i}. \quad (7)$$

**Implementation Details.** For fair comparison with recent works [8, 10, 11, 29, 33], we adopt the reduced ResNet-18 as our feature extraction backbone for all datasets. And the combination of an FC layer and Softmax is employed as the classifier to predict the probability. For model training, an SGD optimizer with a learning rate of 0.1 is used. The batch size is set to 10 for both the data stream and the memory buffer, while the candidate set size for perturbation uncertainty retrieval is 50. To reduce randomness, all reported results are the average of 10 repeated runs.

## 4.2 Performance Comparison

To evaluate the effectiveness of our method, we compare it to several recently published models for OCIL. Since our focus is to enhance replay-based methods, most of these compared approaches are based on memory replay. According to their focus, they can be categorized into 3 types: model update, memory update and memory retrieval. The memory retrieval based methods primarily consider which samples to retrieve for retraining model with new task samples, including MIR [2] and ASER [33]. Methods focusing on memory update mainly consider how to select and store representative samples in buffer, containing GSS [3]. Others focus on model update, including: AGEM [10], ER-WA [45], DER [6], SS-IL [1], DVC [16], AML [8], ACE [8], DER-CBA [38], ACE-MOCA [43]. We also list the performance for fine-tune and iid offline. The former simply updates the model without any strategy against forgetting, acting as the lower bound of continual learning. In the latter, the data of all tasks are available and jointly used for model training, which is regraded as the upper bound. All the reported results are reproduced using their released codes, where we try to select optimal configuration to maximize the final average accuracy.

**Table 1.** Final Average Accuracy on three datasets with different memory sizes. The best scores are in boldface and the second-best results are underlined

Method	Mini-ImageNet			CIFAR-100			CIFAR-10		
	M=1K	M=2K	M=5K	M=1K	M=2K	M=5K	M=0.2K	M=0.5K	M=1K
fine-tune		3.9 ± 0.6			5.7 ± 0.2			17.7 ± 0.4	
iid offline		51.4 ± 0.2			49.6 ± 0.2			81.7 ± 0.1	
AGEM (ICLR' 19)	4.2 ± 0.4	4.4 ± 0.3	4.4 ± 0.2	5.9 ± 0.2	6.0 ± 0.3	5.8 ± 0.3	18.0 ± 0.5	18.2 ± 0.2	18.3 ± 0.2
ER (ICML-W' 19)	10.6 ± 0.6	12.8 ± 1.1	15.4 ± 1.1	11.8 ± 0.4	14.5 ± 0.9	20.1 ± 1.0	23.2 ± 1.0	31.3 ± 1.7	36.7 ± 2.9
GSS (NeurIPS' 19)	10.2 ± 0.7	12.9 ± 1.2	15.4 ± 1.1	10.1 ± 0.5	13.7 ± 0.6	17.4 ± 0.9	22.8 ± 1.3	29.2 ± 1.2	35.3 ± 2.5
MIR (NeurIPS' 19)	10.4 ± 0.7	15.4 ± 1.1	18.7 ± 1.1	11.0 ± 0.4	15.5 ± 0.6	22.1 ± 0.8	23.8 ± 1.2	30.7 ± 2.4	42.8 ± 1.5
GDumb (ECCV' 20)	8.1 ± 0.4	12.4 ± 0.6	20.5 ± 0.6	10.1 ± 0.3	14.4 ± 0.3	20.9 ± 0.3	27.6 ± 1.3	33.1 ± 1.2	38.8 ± 1.1
ER-WA (CVPR' 20)	14.9 ± 0.8	14.7 ± 1	20.3 ± 2.1	16.3 ± 0.6	19.9 ± 0.9	24.3 ± 0.8	28.7 ± 2.2	34.9 ± 2.2	43.6 ± 2.3
DER (NeurIPS' 20)	14.3 ± 0.7	16.7 ± 0.7	15.3 ± 0.7	17.2 ± 0.8	17.9 ± 1.1	18.3 ± 0.7	40.3 ± 2.2	48.2 ± 2.2	52.8 ± 1.4
ASER (AAAI' 21)	12.4 ± 0.8	14.4 ± 1.0	16.3 ± 2.3	14.9 ± 0.6	18.8 ± 0.7	23.5 ± 0.7	30.3 ± 1.4	39.4 ± 1.6	46.6 ± 1.4
SS-IL (ICCV' 21)	15.8 ± 0.8	18.5 ± 0.8	20.0 ± 0.9	18.5 ± 0.8	21.3 ± 0.9	21.9 ± 0.8	36.1 ± 1.3	40.9 ± 2.2	45.5 ± 1.2
DVC (CVPR' 22)	15.4 ± 0.7	17.2 ± 0.8	19.1 ± 0.9	<u>20.6 ± 0.5</u>	<u>22.1 ± 0.9</u>	<u>24.3 ± 0.8</u>	45.4 ± 1.4	50.6 ± 2.9	52.1 ± 2.5
AML (ICLR' 22)	13.6 ± 0.7	15.4 ± 0.6	16.3 ± 0.6	16.3 ± 0.7	16.9 ± 0.9	18.3 ± 0.4	<b>49.3 ± 0.9</b>	<u>53.4 ± 2.7</u>	<u>56.2 ± 2.0</u>
ACE (ICLR' 22)	16.6 ± 0.6	19.5 ± 0.4	21.1 ± 0.5	19.7 ± 0.8	22.1 ± 1.1	23.3 ± 0.6	42.9 ± 1.2	50.1 ± 2.2	53.3 ± 1.6
DER-CBA (ICCV' 23)	18.4 ± 0.8	20.8 ± 0.7	22.0 ± 1.3	19.1 ± 0.8	21.1 ± 0.6	21.2 ± 1.1	41.0 ± 1.5	43.7 ± 2.5	48.7 ± 1.0
ACE-MOCA (TMLR' 23)	<u>21.2 ± 0.7</u>	<u>22.9 ± 0.8</u>	<u>25.0 ± 0.9</u>	19.4 ± 0.8	21.1 ± 0.8	21.6 ± 0.7	45.3 ± 2.2	50.0 ± 2.3	52.4 ± 1.1
Ours	<b>23.4 ± 0.8</b>	<b>25.6 ± 1.3</b>	<b>27.8 ± 0.9</b>	<b>26.5 ± 1.0</b>	<b>29.1 ± 0.6</b>	<b>31.1 ± 0.5</b>	<u>47.8 ± 2.8</u>	<b>56.9 ± 2.3</b>	<b>62.3 ± 1.8</b>

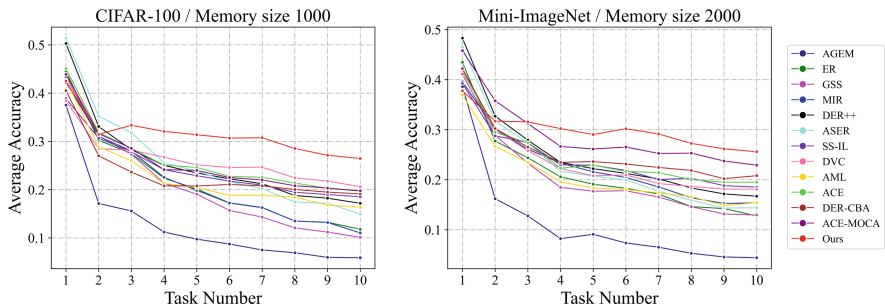
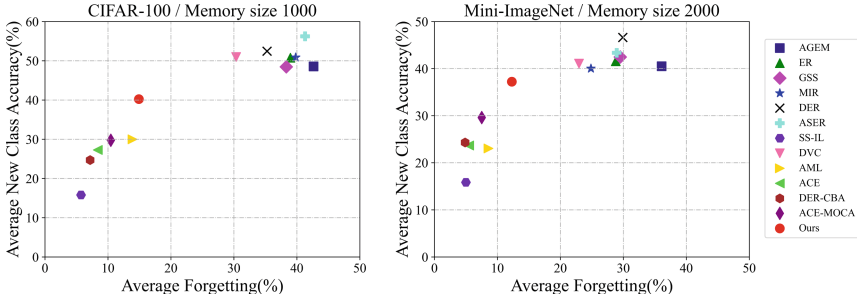
**Fig. 2.** Average accuracy after each incremental learning step.

Table 1 gives the final average accuracy of all methods on three datasets. We can find that our method achieves the best performance in most cases. Specifically, on both Mini-ImageNet and CIFAR-100, our method outperforms all the compared methods by a large margin. For instance, on CIFAR-100, the accuracy increase is 5.9%, 7.0%, and 6.8% respectively for three buffer sizes compared with the second best methods. On CIFAR-10, the gap is still obvious for larger memory. However, when the memory size is 0.2K, the accuracy is slightly inferior to that of AML. In our opinion, the small memory leads to the lower diversity of memory samples, which limits the effect of our method. Moreover, with the growth of memory size, the performance gap becomes larger. For example, the gap on Mini-ImageNet are 2.2%, 2.7% and 2.8% on different memory sizes.

We further show the average accuracy of all observed tasks after each incremental step in Fig. 2. The trend of our method is similar on CIFAR-100 and Mini-ImageNet. Initially, the accuracy is much lower than that of other methods

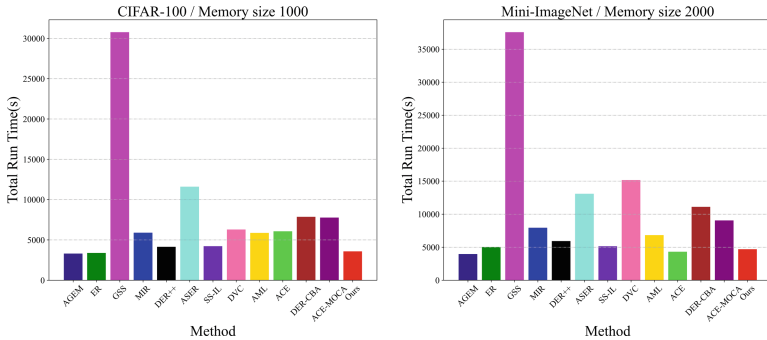


**Fig. 3.** Stability and plasticity trade-off. The closer to the left upper corner, the better the balance is struck.

in the first two tasks. However, from the third task onward, our method consistently obtains the highest accuracy. Compared with our method, although some approaches get higher performance at the beginning, their performance drops dramatically as the number of tasks increases. For example, on the CIFAR-100 and Mini-ImageNet, DER and ASER are optimal at beginning, but their final average accuracy is only at the middle level. This discrepancy is due to the difference between contrastive loss and cross-entropy loss. The former focuses on pulling similar samples closer and separating samples from different classes, while the latter aims at learning classification boundaries. Since the number of classes is small at the beginning, it is relatively easier to learn a clear boundary. Thus, the methods based on CE loss obtain higher accuracy. However, after several incremental learning steps, it becomes more difficult for CE loss to learn a good boundary for many classes. In contrast, contrastive loss can still cluster the feature of each class, which leads to higher performance.

In continual learning, the model needs to learn new knowledge while preserving old knowledge. In other words, the trade-off between stability and plasticity is also very important [44]. To investigate this, we present the interplay between average forgetting and average new class accuracy of different methods in Fig. 3. Obviously, the closer to the left upper corner, the better the balance. In this figure, our method achieves a better stability-plasticity trade-off. Our method exhibits a similar average forgetting rate compared to the methods with much lower forgetting rate, like ACE-MOCA, DER-CBA, ACE, and AML. However, our average new class accuracy is much higher, which means a stronger ability to learn new knowledge.

To compare the computational complexity, we present the running time for all methods in Fig. 4. Although our running time is a little more than AGEM, ER and ER-ACE, our accuracy is much higher than them. Compared to other retrieval methods like MIR, ASER, and DVC, our uncertainty based scoring does not add significant computational cost.



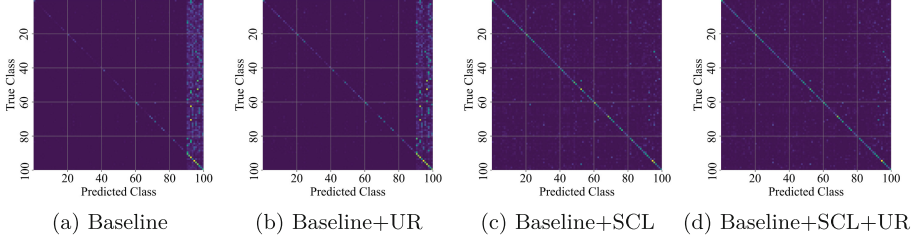
**Fig. 4.** The total running time comparison average over 10 running (including both training time and inference time).

**Table 2.** Influence of different modules on the performance of CIFAR-100.

Memory	M=1k			M=2k			M=5k		
Metrics	FAA $\uparrow$	FAF $\downarrow$	AAN $\uparrow$	FAA $\uparrow$	FAF $\downarrow$	AAN $\uparrow$	FAA $\uparrow$	FAF $\downarrow$	AAN $\uparrow$
Baseline	11.8 $\pm$ 0.4	39.0 $\pm$ 0.9	50.8 $\pm$ 0.8	14.5 $\pm$ 0.9	34.6 $\pm$ 0.8	49.1 $\pm$ 0.5	20.1 $\pm$ 1.0	31.8 $\pm$ 1.1	51.9 $\pm$ 0.8
Baseline + UR	17.4 $\pm$ 0.8	41.5 $\pm$ 0.8	59.1 $\pm$ 0.8	19.1 $\pm$ 0.6	40.2 $\pm$ 1.1	59.7 $\pm$ 0.4	21.0 $\pm$ 1.1	39.7 $\pm$ 1.2	60.5 $\pm$ 0.5
Baseline + SCL	25.5 $\pm$ 0.8	15.7 $\pm$ 1.3	40.8 $\pm$ 1.3	28.3 $\pm$ 1.3	11.5 $\pm$ 2.0	38.9 $\pm$ 3.5	29.6 $\pm$ 1.2	11.1 $\pm$ 2.1	40.2 $\pm$ 1.7
Baseline + SCL + UR	26.5 $\pm$ 1.0	14.9 $\pm$ 2.8	40.2 $\pm$ 3.9	29.1 $\pm$ 0.6	12.3 $\pm$ 1.6	40.1 $\pm$ 3.1	31.1 $\pm$ 0.5	9.8 $\pm$ 2.1	38.9 $\pm$ 3.9

### 4.3 Ablation Study

First, we investigate the impact of various components in our model, whose results are shown in Table 2. The “Baseline” represents the naive ER method. The “UR” denotes the perturbation uncertainty based retrieval strategy while the “SCL” represents the combination of two supervised contrastive learning loss. From this table, we can find that adding any new module can improve the performance dramatically. Specifically, adding UR largely improves the average accuracy of the baseline. Compared to random retrieval, UR selects samples by an interval of  $sz/k$  in the descending order of scores, ensuring that the selected samples include both easy and difficult ones. This retrieval strategy not only increases the diversity of selected samples, but also prevents the model from overemphasizing challenging samples. As a result, it greatly enhances the model’s ability to learn new classes, ultimately leading to a higher average accuracy. In addition, the UR is more effective when the memory size is smaller. The greatest gain is observed when the memory size is  $1k$ . When the memory becomes larger, the class distribution of selected samples by UR is more unbalanced, which will decrease the effect of UR. On the other hand, although adding SCL damages the ability to learn new classes, it greatly reduces the forgetting rate, which finally leads to higher accuracy. In our opinion, the SCL imposes a constraint on the feature distribution of different classes, which helps to learn robust feature. Finally, combining the UR and SCL yields the best performance, which illustrates the complementary of these two modules.



**Fig. 5.** The comparison of confusion matrices on CIFAR-100 with 1k memory.

**Table 3.** Impact of  $\beta$  on the performance of CIFAR-100 with 1k memory.

$\beta$	0.1	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0
Average Accuracy	$26.1 \pm 0.6$	$26.2 \pm 1.0$	$26.5 \pm 1.0$	$25.9 \pm 0.8$	$25.5 \pm 0.9$	$24.9 \pm 0.6$	$24.2 \pm 0.6$	$23.6 \pm 0.8$	$23.0 \pm 0.5$

**Table 4.** Impact of candidate size on performance of CIFAR-100 with 1k and 2k.

M=1K	Candidate Size	30(3.0%)	40(4.0%)	50(5.0%)	100(10%)	200(20%)	300(30%)	400(40%)	500(50%)
	Average Accuracy	$26.3 \pm 1.0$	$26.3 \pm 0.9$	$26.6 \pm 0.7$	$26.0 \pm 1.1$	$26.0 \pm 0.7$	$26.4 \pm 0.6$	$26.4 \pm 0.4$	$26.4 \pm 0.6$
M=2K	Candidate Size	30(1.5%)	40(2.0%)	50(2.5%)	200(10%)	400(20%)	600(30%)	800(40%)	1000(50%)
	Average Accuracy	$28.7 \pm 0.9$	$28.9 \pm 0.7$	$29.1 \pm 0.6$	$29.0 \pm 0.8$	$29.4 \pm 1.0$	$28.5 \pm 0.9$	$28.6 \pm 0.4$	$28.7 \pm 1.0$

To provide a more intuitive insight into the effects of each component, we further compare the confusion matrices of different modules on CIFAR-100 with 1k memory in Fig. 5. From Fig. 5 (a), we can see that ER tends to predict samples to new classes. Adding UR helps the model to achieve more correct classifications (i.e. more diagonal items), as shown in Fig. 5(b). Although UR leads to a few mis-classifications, the task-recency bias is still severe. In contrast, the SCL can largely mitigate task-recency bias and the “Baseline+SCL+UR” achieves the best results according to Fig. 5 (c) and (d).

The influence of  $\beta$  in Eq. (4) on CIFAR-100 is shown in Table 3. The performance initially keeps increasing when using larger  $\beta$ . However, when  $\beta$  is greater than 0.5, the performance decreases. Therefore, we set  $\beta$  to 0.5. Besides, even when  $\beta$  is 0.1, the accuracy is still comparable. We contribute this to the difference of the SS loss and PS loss. Since the samples of old classes are much fewer compared with new classes, the SS loss mainly learns the relationship of classes in the current task. Too large weight for the SS loss will affect model generation. In contrast, the PS loss pulls samples to their corresponding proxy, which is optimized to represent the data of a class.

Finally, we investigate the relation between memory size and candidate size. As shown in Table 4, increasing the size can improve the performance in the beginning. However, when it is larger than 50, using larger candidate size has minimal effect on the performance. Meanwhile, too larger candidate means higher computational cost. Therefore, we set the candidate size as 50 for all datastes and memory sizes, which achieves better performance with lower computation.

## 5 Conclusion

In this work, we propose a simple yet effective method for online class-incremental learning. Specifically, we design perturbation uncertainty based retrieval strategy, which measures the memory samples according to their robustness in the feature space and retrieves samples whose features will be most perturbed. By selecting more meaningful samples, the plasticity of a model can be enhanced. Moreover, we replace the cross-entropy loss with two types of supervised contrastive learning loss. Thus, we can cluster the feature of same class samples and separate the feature of different class, which further enhances the robustness of the features and reduces the task-recency bias. We have conducted extensive experiments on three common benchmarks for OCIL, whose results validate the effectiveness of our model. In the future, we will explore how to select samples with better class distribution and investigate its effect on other continual learning settings.

**Acknowledgments.** This work was supported in part by the National Natural Science Foundation of China (No. 62376218, No. 62101453), Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011298), Natural Science Basic Research Program of Shaanxi (No. 2022JC-DW-08).

## References

1. Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., Moon, T.: Ss-il: separated softmax for incremental learning. In: ICCV, pp. 844–853 (2021)
2. Aljundi, R., et al.: Online continual learning with maximal interfered retrieval. *Adv. Neural Inform. Process. Syst.* **32** (2019)
3. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. *Adv. Neural Inform. Process. Syst.* **32** (2019)
4. Bellitto, G., Pennisi, M., Palazzo, S., Bonicelli, L., Boschini, M., Calderara, S.: Effects of auxiliary knowledge on continual learning. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 1357–1363. IEEE (2022)
5. Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., Calderara, S.: Class-incremental continual learning into the extended der-verse. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(5), 5497–5512 (2022)
6. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. *Adv. Neural. Inf. Process. Syst.* **33**, 15920–15930 (2020)
7. Buzzega, P., Boschini, M., Porrello, A., Calderara, S.: Rethinking experience replay: a bag of tricks for continual learning. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 2180–2187. IEEE (2021)
8. Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., Belilovsky, E.: New insights on reducing abrupt representation change in online continual learning. arXiv preprint [arXiv:2104.05025](https://arxiv.org/abs/2104.05025) (2021)
9. Cha, H., Lee, J., Shin, J.: Co2l: contrastive continual learning. In: ICCV, pp. 9516–9525 (October 2021)
10. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-gem. arXiv preprint [arXiv:1812.00420](https://arxiv.org/abs/1812.00420) (2018)



11. Chaudhry, A., et al.: On tiny episodic memories in continual learning. arXiv preprint [arXiv:1902.10486](https://arxiv.org/abs/1902.10486) (2019)
12. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607. PMLR (2020)
13. Davari, M., Asadi, N., Mudur, S., Aljundi, R., Belilovsky, E.: Probing representation forgetting in supervised and unsupervised continual learning. In: CVPR, pp. 16712–16721 (June 2022)
14. Fu, Z., Wang, Z., Xu, X., Li, D., Yang, H.: Knowledge aggregation networks for class incremental learning. *Pattern Recogn.* **137**, 109310 (2023)
15. Gallardo, J., Hayes, T.L., Kanan, C.: Self-supervised training enhances online continual learning (2021). <https://arxiv.org/abs/2103.14010>
16. Gu, Y., Yang, X., Wei, K., Deng, C.: Not just selection, but exploration: online class-incremental continual learning via dual view consistency. In: CVPR, pp. 7442–7451 (2022)
17. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR, vol. 2, pp. 1735–1742. IEEE (2006)
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR, pp. 9729–9738 (2020)
19. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV, pp. 709–727. Springer (2022)
20. Khosla, P., et al.: Supervised contrastive learning. *Adv. Neural. Inf. Process. Syst.* **33**, 18661–18673 (2020)
21. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**(13), 3521–3526 (2017)
22. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. *Handbook Systemic Autoimmune Diseases* **1**(4) (2009)
23. Li, X., Wang, S., Sun, J., Xu, Z.: Memory efficient data-free distillation for continual learning. *Pattern Recogn.* **144**, 109875 (2023)
24. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(12), 2935–2947 (2017)
25. Liang, G., Chen, Z., Chen, Z., Ji, S., Zhang, Y.: New insights on relieving task-recency bias for online class incremental learning. *IEEE Trans. Circuits Syst. Video Technol.* **34**(5), 3451–3464 (2024)
26. Liu, X., Masana, M., Herranz, L., Van de Weijer, J., Lopez, A.M., Bagdanov, A.D.: Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 2262–2268. IEEE (2018)
27. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. *Adv. Neural Inform. Process. Syst.* **30** (2017)
28. Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., Sanner, S.: Online continual learning in image classification: an empirical survey. *Neurocomputing* **469**, 28–51 (2022)
29. Mai, Z., Li, R., Kim, H., Sanner, S.: Supervised contrastive replay: revisiting the nearest class mean classifier in online class-incremental continual learning. In: CVPR, pp. 3589–3599 (2021)
30. 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.* **45**(5), 5513–5533 (2023)
31. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: the sequential learning problem. In: *Psychology of Learning and Motivation*, vol. 24, pp. 109–165. Elsevier (1989)

32. Shi, F., Wang, P., Shi, Z., Rui, Y.: Selecting useful knowledge from previous tasks for future learning in a single network. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9727–9732. IEEE (2021)
33. Shim, D., Mai, Z., Jeong, J., Sanner, S., Kim, H., Jang, J.: Online class-incremental continual learning with adversarial shapley value. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9630–9638 (2021)
34. Song, K., Liang, G., Chen, Z., Zhang, Y.: Non-exemplar class-incremental learning by random auxiliary classes augmentation and mixed features. *IEEE Trans. Circ. Syst. Video Technol.* (2024)
35. Van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three types of incremental learning. *Nat. Mach. Intell.* **4**(12), 1185–1197 (2022)
36. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. *Adv. Neural Inform. Process. Syst.* **29** (2016)
37. Vitter, J.S.: Random sampling with a reservoir. *ACM Trans. Math. Softw. (TOMS)* **11**(1), 37–57 (1985)
38. Wang, Q., Wang, R., Wu, Y., Jia, X., Meng, D.: Cba: improving online continual learning via continual bias adaptor. In: ICCV, pp. 19082–19092 (2023)
39. Wang, R., et al.: Attriclip: a non-incremental learner for incremental knowledge learning. In: CVPR, pp. 3654–3663 (2023)
40. Wang, Z., et al.: Learning to prompt for continual learning. In: CVPR, pp. 139–149 (2022)
41. Yao, X., et al.: Pcl: proxy-based contrastive learning for domain generalization. In: CVPR, pp. 7097–7107 (2022)
42. Yoon, J., Madaan, D., Yang, E., Hwang, S.J.: Online coreset selection for rehearsal-based continual learning. arXiv preprint [arXiv:2106.01085](https://arxiv.org/abs/2106.01085) (2021)
43. Yu, L., Hu, T., HONG, L., Liu, Z., Weller, A., Liu, W.: Continual learning by modeling intra-class variation. *Trans. Mach. Learn. Res.* (2023). <https://openreview.net/forum?id=iDxfGaMYVr>
44. Zhang, Y., Pfahringer, B., Frank, E., Bifet, A., Lim, N.J.S., Jia, Y.: A simple but strong baseline for online continual learning: Repeated augmented rehearsal. *Adv. Neural. Inf. Process. Syst.* **35**, 14771–14783 (2022)
45. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: CVPR, pp. 13208–13217 (2020)



# Breaking Information Silos: Global Guided Task Prediction for Class-Incremental Learning

Chaoshun Hu<sup>1</sup>, Biaohua Ye<sup>1</sup>, Zixuan Chen<sup>1</sup>, and Jian-Huang Lai<sup>1,2,3</sup>(✉)

<sup>1</sup> School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

{huchsh3,yebh3,chenzx3}@mail2.sysu.edu.cn, stsljh@mail.sysu.edu.cn

<sup>2</sup> Pazhou Lab (Huangpu), Guangzhou 510555, China

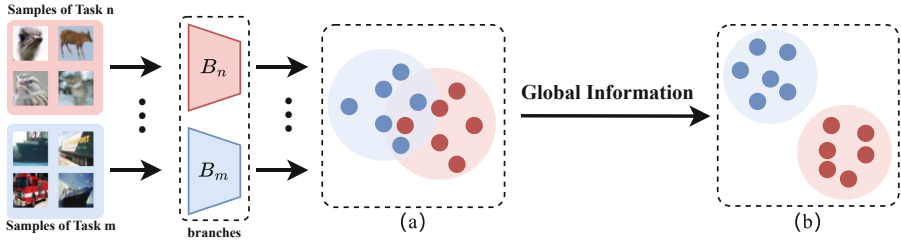
<sup>3</sup> Ministry of Education, Key Laboratory of Machine Intelligence and Advanced Computing, Guangzhou, China

**Abstract.** Class-incremental learning (CIL) aims to learn a series of tasks sequentially, each introducing several new categories. Because providing the task labels during inference can significantly increase accuracy, many approaches attempt to predict task labels in CIL. However, existing works focus on learning local information and overlook the importance of global information. The absence of global guidance leads to the formation of information silos across disparate tasks, resulting in potential inter-task interference. To break information silos, we propose a method called **Global Guided Task Prediction** (GGTP) to introduce global information. Our method consists of two modules. The local de-redundant module aims to reduce information redundancy across different tasks from a global perspective. We combine dual encoders and feature decorrelation loss to effectively reduce redundancy while minimizing catastrophic forgetting. The global information module explicitly extracts global information to serve as auxiliary information guiding task prediction. We extract the most important information from all local information through a global encoder and then aggregate them to form global information. Extensive experiments validate the effectiveness of our method, which achieves state-of-the-art results.

**Keywords:** Class-Incremental Learning · Dynamic Expansion Architectures · Task Prediction

## 1 Introduction

Incremental Learning (IL), an area of machine learning that is evolving quickly due to widespread demand [5, 10, 24], is designed to progressively acquire new classes [35]. Training data is presented as a series of tasks, each introducing a few new classes. Generally speaking, Incremental learning can be roughly divided into two categories of methods: task-incremental learning (TIL) and class-incremental learning (CIL) [17, 34, 41]. The difference between the two lies in whether the task label is known during inference.



**Fig. 1.** (a) Existing methods focus on effectively extracting local information (intra-task information), yet they do not emphasize global information (inter-task information). Without the guidance of global information, information silos may lead to inter-task compactness, thereby affecting the accuracy of task predictions. (b) Global information can enhance inter-task separation of different tasks.

Many researchers [1–3, 15, 22, 23, 36, 42] observe a significant accuracy gap between TIL and CIL and attempt to predict task labels within CIL to reduce the class label search space. Some approaches [1, 3] attempt to predict the task labels through features or gate networks. Others [2, 15, 36, 42] try to predict the task labels by leveraging the data distribution differences across various tasks. Additionally, some methods [22, 23] attempt to introduce Out-of-Distribution Detection classifiers to determine the probability of outlier samples.

However, the above methods mainly focus on local information (intra-task information) and overlook the importance of global information (inter-task information). As illustrated in Fig. 1, we select two tasks,  $m$ , and  $n$  ( $m \neq n$ ), from all tasks, along with their corresponding branches  $B_m$  and  $B_n$ . Since both branches capture local information effectively, both tasks exhibit intra-task compactness. Tasks  $m$  and  $n$  may have some categories with semantic similarities (e.g., cats and dogs), which makes it difficult to distinguish certain samples from the new and old tasks. Different local information forms information silos, lacking connection and updates. Therefore, different tasks may not exhibit inter-task separation. Although many methods do not directly predict task labels based on feature space, but rather on task distribution differences [2, 15, 36, 42] or OOD classifiers [22, 23], the same problems exist. For example, Expert Gate [2] trains an autoencoder for each task and predicts task labels through reconstruction error. Even though each autoencoder can learn local information well, the different autoencoders are information silos. Due to the lack of global information to update local information or to provide auxiliary information, when new samples are input into old autoencoders, they may exhibit reconstruction errors of a similar magnitude to old samples, leading to incorrect task prediction.

To break the information silos between tasks, we propose a method named Global Guided Task Prediction (GGTP) to introduce global information in task prediction. We argue that global information can better guide the learning of local information and also directly guide task prediction. Therefore, we propose two modules: the Local De-redundant Module and the Global Information Mod-

ule. The **Local De-redundant Module** aims to reduce inter-task redundancy from a global perspective to increase task separability. We utilize a dual-task encoder mechanism to allow local information to update gradually, reducing catastrophic forgetting. Next, we introduce a feature decorrelation loss to eliminate redundancy between different tasks. The **Global Information Module** aims to leverage global information to directly guide task predictions. The local information is first purified through a global encoder and then aggregated to get global information. Global information can better represent the relationships between different tasks and guide tasks to exhibit better separability. As a result, the global information captured by the proposed modules can guide the model for accurate task prediction. Extensive experiments conducted on widely used datasets CIFAR100 [27] and ImageNet [8] demonstrate that our method effectively surpasses state-of-the-art performance.

Contributions of this paper can be summarized as follows:

1. We propose GGTP to incorporate global information into task prediction to break the information silos between tasks.
2. Our proposed Local De-redundant Module reduces redundant information between tasks. Our proposed Global Information Module extracts global information to better guide task prediction.
3. Significant performance gains demonstrated by extensive experiments on CIFAR100, ImageNet datasets.

## 2 Related Work

Class-incremental learning aims to balance the contradiction between stability and plasticity. A pioneering method in this area is iCaRL [35], which uses a greedy strategy to retain key samples that are most helpful in reducing forgetting. Building on this, methods like WA [53], UCIR [18], BiC [47], and MAFRC [6] recognize that an imbalance between new and old classes leads to biases in classifiers. Some approaches [4, 32, 39, 50] impose constraints on the gradient update of new class samples, ensuring that the update direction forms an acute angle or is orthogonal to the gradient of old class samples, thereby not increasing the loss of old classes. Some methods [9, 20, 29, 38, 47, 52] impose constraints on the features or logits of the model’s output. Some works [25, 28, 49, 51] also recognize that different parameters in the model contribute unequally to retaining old knowledge, thus proposing several methods to estimate the importance of each parameter and freezing the most critical parameters to preserve existing knowledge. Next, we will introduce two types of methods that are most relevant to the work in this paper.

**Dynamically Expansion Architecture.** Previous work inevitably faces an issue: the limited old class samples could not represent the complete data distribution. The model may overfit these few samples when it updates, causing information loss. To address this, Yan et al. [48] introduce the concept of dynamic

expansion architecture. This approach preserves existing knowledge by freezing the existing feature extractors and adding a new extractor to learn features of new classes. Huang et al. [19] incorporate attention mechanisms and multi-level knowledge distillation, aiming to reduce old-new confusion. Wang et al. [43] achieve bidirectional compatibility, where the module responsible for a given task would dominate the prediction under ideal circumstances. With each new task, the model adds extra parameters. This constant increase in parameters limits the practicality of these methods. To mitigate this, Wang et al. [44] employ knowledge distillation [16] after each task to produce a student model comparable in size to the original, thus addressing the issue of parameter inflation.

**Task Prediction.** iTAML [36] processes test samples in batches when predicting tasks. Each batch must come from a single task. This limits the practicality of the method. Expert Gate [2] builds a gating autoencoder for each task. It determines the best matching task by comparing the difference between the reconstructed sample and the original sample. Davide et al. [1] design an independent feature extractor for each task using a gating scheme. It requires only a small number of parameters to add a new feature extractor. After extracting features, this method concatenates them and sends them to a task predictor for prediction. HyperNet [42] and PMCL [15] propose an entropy-based task-id prediction method. MORE [23] and CLOM [22] incorporate Out-of-Distribution Detection (OOD) to determine whether the current sample belongs to the task. Cai et al. [3] design a gate network in conjunction with a dynamic expansion architecture to predict the task label of a sample during inference, reducing inter-task confusion. However, this method primarily uses the dynamic expansion architecture as a strong baseline and does not effectively leverage the advantages of multi-branches.

**Global Information.** DKT [13] introduces a task general token to store global information (task-general knowledge). Similar to Prefix Tuning [30], the task general token is concatenated with the input tokens and used as  $K$  and  $V$  inputs to the attention block. Without constraints to protect old knowledge, the task general token forgets old global knowledge when learning new tasks, resulting in biased global cognition. In contrast, our proposed Global Information Module generates global information by purifying and aggregating all task-specific information, resulting in more comprehensive global information. Based on L2P [46], DualPrompt [45] subdivides prompts into G-prompts and E-prompts, which store task-invariant instructions and task-specific instructions, respectively. Similar to DKT, DualPrompt also lacks constraints to protect old knowledge, leading to G-prompts forgetting old task-invariant information. VIDA [31] finds that high-rank and low-rank adapters can represent domain-shared and domain-specific knowledge, respectively. Therefore, it decouples them using high and low-rank adapters and dynamically merges these types of knowledge through the Homeostatic Knowledge Allotment strategy. The domain-shared information in VIDA represents the common information across different domains, whereas

our global information represents the important information that distinguishes different tasks.

### 3 Methods

#### 3.1 Problem Setting and Method Overview

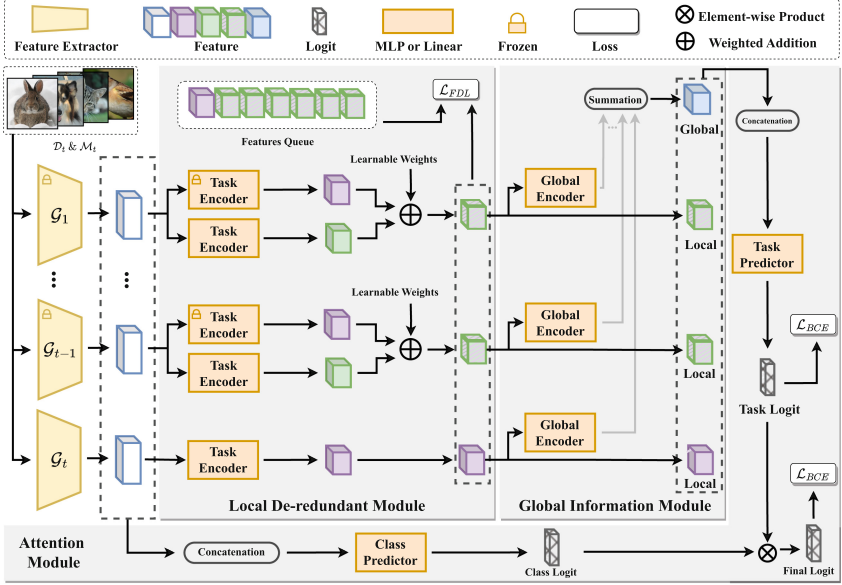
**Problem Setting.** First, we introduce the setup of class-incremental learning. In this setup, a model sequentially obtains a series of datasets  $\{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T\}$ . Each dataset contains classes that the model has not learned before. The process of learning dataset  $\mathcal{D}_t$  is referred to as the  $t$  step. All classes and their corresponding samples in  $\mathcal{D}_t$  constitute the  $t$  task. After each step, the model is tested on all classes learned so far.

**Method Overview.** Next, we will outline the framework and notation of our method. In the first step, the feature extractor  $\mathcal{G}_1$  and classifier  $\mathcal{F}_1$  are trained the same as the general classification task on dataset  $\mathcal{D}_1$ . At each subsequent step  $t \in \{2, 3, \dots, T\}$ , we freeze all the feature extractors  $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{t-1}\}$ . We then add a new feature extractor  $\mathcal{G}_t$ , a new classifier  $\mathcal{F}_t$ , and an auxiliary classifier  $\mathcal{F}_t^a$ . The auxiliary classifier  $\mathcal{F}_t^a$  is only used during step  $t$  and is not retained for the next step. During training and inference, a sample is fed into all feature extractors, resulting in  $t$  feature  $\{\phi_1, \phi_2, \dots, \phi_t\}$ . Then concatenate all the features into one feature  $\phi$  by dimension. Shown in Fig. 2, our model consists of three components: Local De-redundant Module Module, Global Information Module, and Attention Module. We introduce the three modules in turn and then finish with the optimizing and lightweight. The encoder we use in these modules is a linear layer, except for the global encoder, which is a two-layer MLP, so they do not significantly increase the model’s complexity.

#### 3.2 Local De-Redundant Module

**Dual Task Encoder Mechanism.** The feature extractor primarily learns to distinguish between different classes, which may result in features that are not well-suited for differentiating between tasks. To address this, we obtain local features by processing the original features through task encoders. These local features have a lower dimensionality, which reduces the number of parameters across all encoders.

For the new task, features  $\phi_t$  are processed through a task encoder  $E_t^S$  to acquire local features. For the old tasks, to reduce the catastrophic forgetting associated with updating local information, we introduce two task encoders for each old task. The first encoder,  $\phi_m^S$ , is inherited from the previous step. The second encoder,  $\phi_m^P$ , is created in the current step and is initialized with  $\phi_m^S$ . To protect old knowledge, we freeze the set of encoders  $\{E_1^S, E_2^S, \dots, E_{t-1}^S\}$ . At the same time, we update the set  $\{E_1^P, E_2^P, \dots, E_{t-1}^P\}$  to acquire new knowledge. We then generate a learnable feature weight vector  $\alpha_m$  for each task. The dimensions



**Fig. 2.** Schematic diagram of our method. First, samples are passed through a feature extractor to extract diverse local information. Then, a Local De-redundant Module removes the redundancy of local information. This is followed by a Global Information Module that extracts global information. Finally, an Attention Module fuses the results of task prediction with the results of class prediction.

of  $\alpha_m$  match those of the local features. By default, we initialize  $\alpha_m$  to 1, which biases the fused features towards retaining their original features. We first extract all the local features:

$$\begin{aligned}\phi_m^P &= E_m^P(\phi_m), m = 1, 2, \dots, t-1 \\ \phi_m^S &= E_m^S(\phi_m), m = 1, 2, \dots, t-1\end{aligned}\quad (1)$$

The feature weight vector  $\alpha_m$  is passed through a sigmoid function to obtain a coefficient vector that ranges between 0 and 1. Subsequently, we calculate a weighted sum of  $\phi_m^P$  and  $\phi_m^S$  using this  $\alpha_m$  to produce the fused feature  $\phi_m^T$ . Here,  $\sigma$  represents the sigmoid function:

$$\phi_m^T = \sigma(\alpha_m) \cdot \phi_m^S + (1 - \sigma(\alpha_m)) \cdot \phi_m^P \quad (2)$$

So the final  $\phi_m^T$  can be expressed as:

$$\phi_m^T = \begin{cases} \sigma(\alpha_m) \cdot \phi_m^S + (1 - \sigma(\alpha_m)) \cdot \phi_m^P & \text{if } m = 1, 2, \dots, t-1 \\ \phi_m^S & \text{if } m = t \end{cases} \quad (3)$$



At the end of each step, assuming the weights for  $E_m^S$  and  $E_m^P$  are  $\mathcal{W}_m^S$  and  $\mathcal{W}_m^P$  respectively, the final output  $\phi_m^T (m < t)$  can be expressed as follows:

$$\begin{aligned}\phi_m^T &= \sigma(\alpha_m) \cdot \mathcal{W}_m^S \cdot \phi_m + (1 - \sigma(\alpha_m)) \cdot \mathcal{W}_m^P \cdot \phi_m \\ &= \left( \sigma(\alpha_m) \cdot \mathcal{W}_m^S + (1 - \sigma(\alpha_m)) \cdot \mathcal{W}_m^P \right) \cdot \phi_m\end{aligned}\quad (4)$$

Thus, we update  $E_m^S$  and discard  $E_m^P$ :

$$\mathcal{W}_m^S \leftarrow \sigma(\alpha_m) \cdot \mathcal{W}_m^S + (1 - \sigma(\alpha_m)) \cdot \mathcal{W}_m^P \quad (5)$$

So we mitigate catastrophic forgetting by gradually updating old knowledge.

**Feature Decorrelation Loss.** We propose a feature decorrelation loss to eliminate the redundancy of different tasks. We treat each feature dimension as a random variable. Assuming the dimension of  $\phi_m^T$  is  $d$ , there are a total of  $t \cdot d$  random variables. We encourage the covariance of these random variables to be as close to zero as possible. Direct estimation of covariance may not be accurate due to the limited number of old class samples. To address this issue, we design a feature queue  $\mathcal{Q}$  to store a certain number of old class features, with the queue length set to  $q$ . For each batch of  $b$  input samples, we combine the features in the queue  $\mathcal{Q}$  with features of the current batch. Let  $\varphi_k$  represent the  $k$ -th dimension of the feature  $\phi^T$ , then the feature decorrelation loss is defined as follows:

$$\mathcal{L}_{FDL} = \frac{1}{(t \cdot d)^2 - t \cdot d} \sum_{k=1}^{t \cdot d} \sum_{\substack{m=1 \\ m \neq k}}^{t \cdot d} |\text{cov}(\varphi^k, \varphi^m)|, \quad (6)$$

where  $\text{cov}$  denotes the calculation of covariance based on  $b+q$  values within each batch, and  $|a|$  denotes the absolute value. At the end of each batch, we add the features of the old classes from the current batch to the queue  $\mathcal{Q}$  and remove an equal number of the earliest features that entered the queue. This ensures that the size of the queue remains and the queue stores the most recent small portion of old class features.

### 3.3 Global Information Module

Global information can articulate the relationships between different tasks and serve as additional guidance for task prediction. Even though local information may already imply the relationships among various tasks, local information is often rich and may obscure critical details. For instance, consider a scenario where one task involves red targets while another involves green targets. The local information could contain a plethora of details, such as shape or texture, but the key distinction between the tasks might simply be the color difference. All local information is processed through a global encoder  $E_m^G$  to filter out irrelevant details, and then aggregated to form the final global information:

$$\phi^G = \sum_{m=1}^t E_m^G(\phi_m^T) \quad (7)$$

### 3.4 Attention Module

Task confidence is used as the weight for an attention mechanism to either amplify or diminish the probabilities of different categories. First, We combine all local features,  $\phi_m^T$ , with the global feature,  $\phi^G$ , along their dimensions. This results in the final task feature,  $\phi^T$ . Then, we obtain the class logit  $l^c$  through the classifier  $\mathcal{F}_t$  and the task logit  $l^t$  through the task predictor  $\mathcal{F}_t^{task}$ . To maintain numerical stability,  $l^t$  is scaled to the range of  $[0, 1]$  using the Sigmoid function.  $l^t$  is then expanded to match the length of  $l^c$  based on the number of categories per task. Finally, we combine these two logits by element-wise multiplication to produce the final logit  $l^f$ . Thus, the final logit  $l^f$  takes into account both task and category information. Information from task predictions effectively increases the probability of the corresponding task while decreasing the probability of other tasks.

### 3.5 Optimizing and Lightweight Model

**Optimizing.** Like DER [48], in the first stage, the model focuses on learning the features of new tasks. The second stage is dedicated to training the classifier. During the first stage, we train the features  $\phi_t$  using an auxiliary classifier  $\mathcal{F}_t^a$ . Unlike DER, the logit output by our auxiliary classification head includes all classes, not just past classes grouped as a single pseudo-class. This is because one pseudo-class can only express features common to old classes, which is not conducive to the new feature extractor learning to distinguish between new classes and specific similar old ones. This helps the new feature extractor learn the characteristics of the new task. In the second stage, to address the imbalance between new and old classes, we train the task logits  $l^t$  and final logits  $l^f$  using a balanced cross-entropy loss [37].

**Lightweight Model.** Similar to [44], we reduce the model’s parameter count using knowledge distillation. After each step, we use the current model as a teacher and employ knowledge distillation to create a single-branch student model and discard the teacher model. Since the model has at most two branches, we treat all old classes as a single task, and each feature extractor is equipped with just one task encoder. We name this streamlined model GGTP-Lite. To address the imbalance between new and old classes, we propose a balanced knowledge distillation function inspired by [37]. The logits from the teacher and student models are denoted as  $l^t$  and  $l^s$ , respectively.  $l_i^t$  and  $l_i^s$  represent the outputs for the  $i^{th}$  dimension. The variable  $\eta_i$  indicates the number of samples for the  $i^{th}$  class. The loss formula is as follows:

$$\mathcal{L}_{BKD} = - \sum_i \frac{\exp(l_i^t/T)}{\sum_j \exp(l_j^t/T)} \cdot \log \left( \frac{\exp((l_i^s + \log(\eta_i))/T)}{\sum_j \exp((l_j^s + \log(\eta_j))/T)} \right) \quad (8)$$

Compared to FOSTER [44], Our proposed  $\mathcal{L}_{BKD}$  has no additional hyperparameters.

**Table 1.** Test results on CIFAR-100. #P means the number of parameters after completing the learning of all tasks. Avg means the average accuracy (%) over steps.

Methods	Pub	CIFAR100-B0						CIFAR100-B50			
		5 Step		10 Step		20 Step		5 Step		10 Step	
		#P	Avg	#P	Avg	#P	Avg	#P	Avg	#P	Avg
iCaRL [35]	CVPR'17	11.2	71.14	11.2	61.20	11.2	61.20	11.2	65.06	11.2	71.14
UCIR [18]	CVPR'19	11.2	62.77	11.2	58.17	11.2	58.17	11.2	64.28	11.2	62.77
BiC [47]	CVPR'19	11.2	73.10	11.2	66.48	11.2	66.48	11.2	66.62	11.2	73.10
WA [53]	CVPR'20	11.2	72.81	11.2	67.33	11.2	67.33	11.2	64.01	11.2	72.81
PODNet [11]	ECCV'20	11.2	66.70	11.2	53.97	11.2	53.97	11.2	67.25	11.2	66.70
DyTox [12]	CVPR'22	10.7	73.66	10.7	67.30	10.7	67.30	-	-	-	-
FOSTER [44]	ECCV'22	11.2	77.61	11.2	75.18	11.2	72.26	11.2	75.11	11.2	70.21
MAFDRC [6]	ICCV'23	11.2	78.70	11.2	76.93	11.2	74.09	11.2	74.95	11.2	72.26
GGTP-Lite	Ours	11.2	<b>80.28</b>	11.2	<b>79.51</b>	11.2	<b>77.23</b>	11.2	<b>77.95</b>	11.2	<b>75.56</b>
DER [48]	CVPR'21	56.0	79.03	112	78.13	224	77.85	67.2	77.15	123.2	75.58
MCTD [3]	CVPR'23	67.2	78.15	123.2	77.40	235.2	76.20	78.4	76.19	134.4	75.43
TCIL [19]	AAAI'23	56.0	80.23	112	79.12	224	78.10	67.2	77.76	123.2	<b>76.91</b>
GGTP	Ours	57.6	<b>81.21</b>	115.2	<b>80.73</b>	230.4	<b>79.61</b>	69.1	<b>78.66</b>	126.7	76.53

## 4 Experiments

### 4.1 Experiment Setup and Implementation Details

**Dataset.** We evaluated on the widely used incremental learning dataset CIFAR-100 [27] and ImageNet100/1000 [8]. CIFAR-100 includes 100 classes, with 500 training images for each class and 100 evaluation images for each class, with a resolution of  $32 \times 32$ . ImageNet-1000 is a large-scale dataset consisting of 1000 classes with a total of 1.28 million training images and 500 test images per class. ImageNet-100 is a dataset composed of 100 randomly selected classes from ImageNet-1000. The sequence numbers of the 100 classes we selected are derived from [35].

**Evaluation Protocols.** Following [35], we evaluated five widely used testing protocols. (1) CIFAR100-B0: Divide the 100 classes of CIFAR100 equally into 5, 10, and 20 tasks, with a fixed memory budget of 2000 samples. (2) CIFAR100-B50: Pretrain the 50 classes of CIFAR100, then divide the remaining 50 classes equally into 5 and 10 tasks, and retain 20 samples for each class. (3) ImageNet100-B0: Divide the 100 classes of ImageNet100 equally into 10 tasks, with a fixed memory budget of 2000 samples. (4) ImageNet100-B50: Pretrain the 50 classes of ImageNet100, then divide the remaining 50 classes equally into 10 tasks, and retain 20 samples for each class. (5) ImageNet1000-B0: Divide the 1000 classes of ImageNet1000 equally into 10 tasks, with a fixed memory budget of 20000 samples. The class increment order also follows [35], using 1993 as the seed to generate the increment class order.

**Table 2.** Test results on ImageNet-100 and ImageNet-1000.

Methods	Pub	ImageNet100-B50		ImageNet100-B0		ImageNet1000-B0	
		#P	Avg	#P	Avg	#P	Avg
UCIR [18]	CVPR'19	11.2	68.09	–	–	–	–
PODNet [11]	ECCV'20	11.2	74.33	–	–	–	–
TPCIL [40]	ECCV'20	11.2	74.81	–	–	–	–
FOSTER [44]	ECCV'22	11.2	77.54	11.2	78.40	11.2	68.34
MAFDRC [6]	ICCV'23	11.2	77.95	11.2	79.66	11.2	69.37
GGTP-Lite	Ours	11.2	<b>78.34</b>	11.2	<b>80.05</b>	11.2	<b>69.72</b>
DER [48]	CVPR'21	123.2	77.74	112	79.81	112	69.81
MCTD [3]	CVPR'23	134.4	79.83	123.2	80.46	123.2	70.08
TCIL [19]	AAAI'23	–	–	112	77.66	–	–
BEEF-C [43]	ICLR'23	–	–	112	79.34	–	–
GGTP	Ours	126.1	<b>80.89</b>	114.6	<b>82.03</b>	119.2	<b>72.60</b>

**Implementation Details.** For CIFAR-100, we utilize an adjusted ResNet-18 [48] as the feature extractor and set the batch size to 128. For ImageNet, we employ the standard ResNet-18 [14] as the feature extractor with a batch size of 256. For both CIFAR-100 and ImageNet, during the first stage, we set the initial learning rate to 0.1 and use a cosine annealing scheduler to decay the learning rate to 0 over the epochs. In the second stage, all other settings for the learning rate remain the same, except that the initial learning rate is set to 0.001. We use SGD with a momentum of 0.9 and a weight decay of  $5e-4$ . The size of the feature queue  $\mathcal{Q}$  is set to 300. The coefficient for feature decorrelation loss is set to 1. The output dimension of the task encoder is set to one-quarter of the input dimension. Follow [3, 6, 19, 44], for data augmentation, we uniformly use [7], random cropping, horizontal flip, and normalization.

## 4.2 Results and Discussion

**Result on CIFAR-100.** Shown in Table 1, we compare with many methods, such as iCaRL [35], UCIR [18], BiC [47], WA [53], PODNet [11], DyTox [12], MAFDRC [6], DER [48], MCTD [3], TCIL [19] and FOSTER [44]. The module we propose adds a very small number of parameters. The increase in parameters is about 2.9%. Because #P means the number of parameters after completing the learning of all tasks, GGTP-Lite has the same number of parameters as other single-branch models. Overall, our method is only slightly behind TCIL on CIFAR100-B50S10 but outperforms the best results under the other four protocols. On CIFAR100-B0S5, CIFAR100-B0S10, CIFAR100-B0S20, and CIFAR100-B50S5, our results exceed the best results by **0.98%**, **1.61%**, **1.51%**, and **0.90%**, respectively. It is noteworthy that our model maintains a rapid inference speed, as we do not modify the backbone or introduce additional huge

gating networks, unlike other task-prediction-based methods. The overall performance of GGTP-Lite is also quite good, even outperforming many multi-branch dynamic extension architecture methods on some protocols.

**Result on ImageNet.** Shown in Table 2, for ImageNet-100 and ImageNet-1000, we compare with UCIR [18], PODNet [11], TPCIL [40], DER [48], MCTD [19], MAFDRC [6], MCTD [3], TCIL [19], BEEF [43], DER [48] and FOSTER [44]. Overall, our method significantly outperforms all other approaches on ImageNet. On ImageNet100-B50, ImageNet100-B0, and ImageNet1000-B0, our method surpasses the best results byyu, respectively. This may suggest that our method is more effective on larger datasets. The test results on ImageNet1000 indicate that the model we propose is better suited to adapt to large-scale datasets.

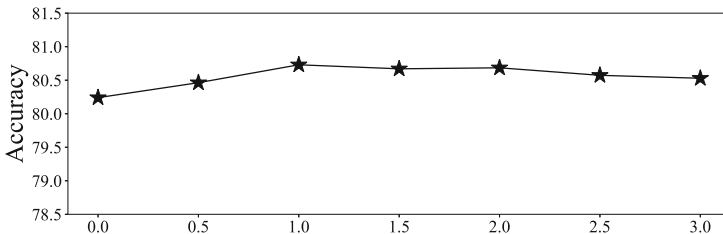
### 4.3 Ablation Study and Analysis

The following experiments are all based on the CIFAR100-B0S10 protocol.

**Table 3.** Ablation study results for various components.

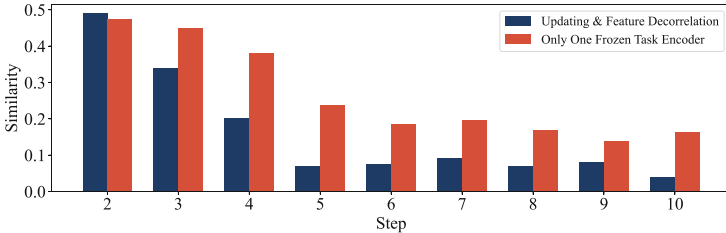
Baseline	Dual Task Encoder	Feature Decorrelation	Global Information	Avg	Last
✓				78.13	68.82
✓	✓			79.31	70.76
✓	✓	✓		80.15	71.62
✓	✓	✓	✓	80.73	72.86

**Different Components in Ours Method.** We conduct ablation experiments on the various components we propose, shown in Table 3. Our module primarily consists of four parts: Baseline (DER), Dual Task Encoder, Feature Decorrelation Loss, and Global Information. In the table, these are respectively denoted as Baseline, Dual Task Encoder, Feature Decorrelation, and Global Information. The “Last” column indicates the overall accuracy across all categories after the model has learned all tasks. Our components contributed to accuracy improvements of **1.94%**, **0.86%**, and **1.24%** respectively.



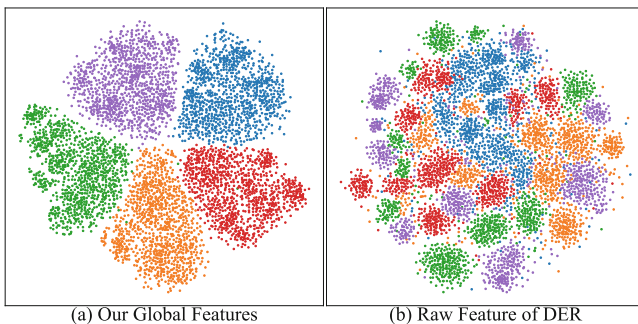
**Fig. 3.** Coefficient of the  $\mathcal{L}_{FDL}$ .

**Sensitivity Analysis of Hyper-Parameters.** Next, we perform a sensitivity analysis on the coefficient of the feature decorrelation loss, as illustrated in Fig. 3. We test it in the interval  $[0, 3]$  with increments of 0.5. Within this range, accuracy exhibits a trend of initially increasing and then decreasing. The overall trend is relatively flat, with an optimal value of 1.



**Fig. 4.** Results of centered Kernel alignment (CKA) between different  $\phi_m^T$ .

**Similarity of Local Features.** Following [21], we utilize the Centered Kernel Alignment (CKA) [26] to measure the similarity between different local information, with higher values indicating greater similarity. To validate the effect of our proposed Local De-redundant Module, we retrain a “stability” model. This model generates only a new  $E_t^S$  at each step without producing any  $E_t^P$ . As a result, the earlier modules cannot update their knowledge in subsequent steps. As shown in Fig. 4, we calculated and averaged the similarity between different  $\phi_m^T$  at the end of each step. The findings indicate that our improved model effectively reduces the similarity and enhances the differences between different local information, compared to the “stability” model.



**Fig. 5.** t-SNE [33] visualisation results of features from different tasks.

**Visualisation Results of Tasks.** We conducted a t-SNE [33] visualization analysis for the raw feature  $\phi$  of DER and our proposed global feature  $\phi^G$ , samples from different tasks with distinct colors, as shown in Fig. 5. The visualization reveals that the feature  $\phi$  focuses more on distinguishing between categories

rather than tasks, with samples from various tasks not converging together. In contrast, global Information  $\phi^G$  eliminates a large amount of irrelevant information, articulating the relationships between different tasks. Therefore, it can effectively guide task prediction.

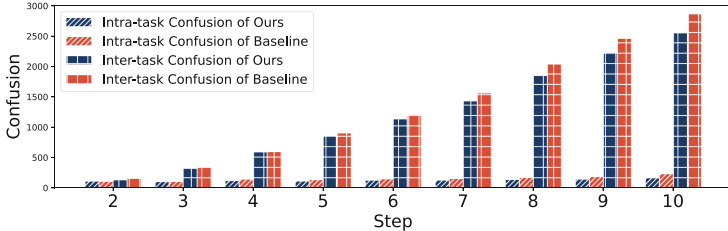


Fig. 6. Confusion analysis on the baseline and ours.

**Confusion Analysis.** Confusion can be divided into intra-task confusion and inter-task confusion. Task prediction can effectively solve inter-task confusion. To verify this, we conduct a confusion analysis on the baseline and our improved model. The results are shown in Fig. 6. The analysis reveals that confusion between tasks sharply increases, while confusion within tasks grows slowly. After integrating our modules, inter-task confusion significantly decreases. This leads to a substantial improvement in overall classification accuracy.

## 5 Conclusion

In this paper, we propose a method called **Global Guided Task Prediction (GGTP)** to introduce global information to solve the problem of information silos in task prediction. Our method consists of two modules. The **Local Redundant Module** removes redundancy from different local information. In particular, it updates local information on old tasks while reducing catastrophic forgetting. The **Global Information Module** distills important information from all local information to express the relationships between different tasks and then serves as auxiliary information to guide task prediction. Our experiments confirm that this method effectively predicts task labels and achieves state-of-the-art performance. Our method relies on local information to reduce catastrophic forgetting of global information. This means that the model must retain all the local information, leading to an increase in the number of parameters as tasks increase. Exploring how to directly update global information without relying on local information, while simultaneously reducing the forgetting of learned global information, is a potential research direction.

**Acknowledgement.** This work was supported in part by the Key Areas Research and Development Program of Guangzhou under Grant 2023B01J0029; and in part by the Key Area Research and Development Program of Guangdong Province, China, under Grant 2018B010109007; and in part by the National Natural Science Foundation of China under Grant 62076258.

## References

1. Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., Bejnordi, B.E.: Conditional channel gated networks for task-aware continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3931–3940 (2020)
2. Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3366–3375 (2017)
3. Cai, T., et al.: Multi-centroid task descriptor for dynamic class incremental inference. In: CVPR, pp. 7298–7307 (2023)
4. Chaudhry, A., Ranzato, M., Rohrbach, M., Elhoseiny, M.: Efficient lifelong learning with a-gem. arXiv preprint [arXiv:1812.00420](https://arxiv.org/abs/1812.00420) (2018)
5. Chen, H., Zhang, Q., Lai, J.H., Xie, X.: Unsupervised group re-identification via adaptive clustering-driven progressive learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 1054–1062 (2024)
6. Chen, X., Chang, X.: Dynamic residual classifier for class incremental learning. In: ICCV, pp. 18743–18752 (2023)
7. Cubuk, E., Zoph, B., Mane, D., Vasudevan, V., Le, Q.: Autoaugment: learning augmentation strategies from data. In: CVPR, pp. 113–123 (2019)
8. 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)
9. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5138–5146 (2019)
10. Ding, G., Golong, H., Yao, A.: Coherent temporal synthesis for incremental action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 28485–28494 (2024)
11. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: PODNet: pooled outputs distillation for small-tasks incremental learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12365, pp. 86–102. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58565-5\\_6](https://doi.org/10.1007/978-3-030-58565-5_6)
12. Douillard, A., Ramé, A., Couairon, G., Cord, M.: Dyttox: Transformers for continual learning with dynamic token expansion: Supplementary materials
13. Gao, X., He, Y., Dong, S., Cheng, J., Wei, X., Gong, Y.: Dkt: diverse knowledge transfer transformer for class incremental learning. In: CVPR, pp. 24236–24245 (2023)
14. 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)
15. Henning, C., et al.: Posterior meta-replay for continual learning. *Adv. Neural. Inf. Process. Syst.* **34**, 14135–14149 (2021)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv (2015)
17. Hossain, M.S., Saha, P., Chowdhury, T.F., Rahman, S., Rahman, F., Mohammed, N.: Rethinking task-incremental learning baselines. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 2771–2777. IEEE (2022)
18. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: CVPR, pp. 831–839 (2019)



19. Huang, B., Chen, Z., Zhou, P., Chen, J., Wu, Z.: Resolving task confusion in dynamic expansion architectures for class incremental learning. In: AAAI, vol. 37, pp. 908–916 (2023)
20. Kang, M., Park, J., Han, B.: Class-incremental learning by knowledge distillation with adaptive feature consolidation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16071–16080 (2022)
21. Kim, D., Han, B.: On the stability-plasticity dilemma of class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20196–20204 (2023)
22. Kim, G., Esmailpour, S., Xiao, C., Liu, B.: Continual learning based on ood detection and task masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3856–3866 (2022)
23. Kim, G., Liu, B., Ke, Z.: A multi-head model for continual learning via out-of-distribution replay. In: Conference on Lifelong Learning Agents, pp. 548–563. PMLR (2022)
24. Kim, J., Cho, H., Kim, J., Tiruneh, Y.Y., Baek, S.: Sddgr: stable diffusion-based deep generative replay for class incremental object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 28772–28781 (2024)
25. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
26. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning, pp. 3519–3529. PMLR (2019)
27. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
28. Lee, J., Hong, H.G., Joo, D., Kim, J.: Continual learning with extended kronecker-factored approximate curvature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9001–9010 (2020)
29. Lee, K., Lee, K., Shin, J., Lee, H.: Overcoming catastrophic forgetting with unlabeled data in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 312–321 (2019)
30. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint [arXiv:2101.00190](https://arxiv.org/abs/2101.00190) (2021)
31. Liu, J., et al.: Vida: Homeostatic visual domain adapter for continual test time adaptation. arXiv preprint [arXiv:2306.04344](https://arxiv.org/abs/2306.04344) (2023)
32. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. Advances in neural information processing systems **30** (2017)
33. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR **9**(11) (2008)
34. Pernici, F., Bruni, M., Bacchi, C., Turchini, F., Del Bimbo, A.: Class-incremental learning with pre-allocated fixed classifiers. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 6259–6266. IEEE (2021)
35. Rebuffi, S.-A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR, pp. 2001–2010 (2017)
36. Rajasegaran, J., Khan, S., Hayat, M., Khan, F.S., Shah, M.: itaml: an incremental task-agnostic meta-learning approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13588–13597 (2020)
37. Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al.: Balanced meta-softmax for long-tailed visual recognition. Adv. Neural. Inf. Process. Syst. **33**, 4175–4186 (2020)

38. Smith, J., Hsu, Y.C., Balloch, J., Shen, Y., Jin, H., Kira, Z.: Always be dreaming: a new approach for data-free class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9374–9384 (2021)
39. Tang, S., Chen, D., Zhu, J., Yu, S., Ouyang, W.: Layerwise optimization by gradient decomposition for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9634–9643 (2021)
40. Tao, X., Chang, X., Hong, X., Wei, X., Gong, Y.: Topology-preserving class-incremental learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 254–270. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58529-7\\_16](https://doi.org/10.1007/978-3-030-58529-7_16)
41. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint [arXiv:1904.07734](https://arxiv.org/abs/1904.07734) (2019)
42. Von Oswald, J., Henning, C., Grewe, B.F., Sacramento, J.: Continual learning with hypernetworks. arXiv preprint [arXiv:1906.00695](https://arxiv.org/abs/1906.00695) (2019)
43. Wang, F., et al.: Beef: bi-compatible class-incremental learning via energy-based expansion and fusion. In: ICLR (2022)
44. Wang, F., Zhou, D., Ye, H., Zhan, D.: Foster: Feature boosting and compression for class-incremental learning. In: ECCV. pp. 398–414. Springer (2022)
45. Wang, Z., et al.: Dualprompt: complementary prompting for rehearsal-free continual learning. In: ECCV, pp. 631–648. Springer (2022). [https://doi.org/10.1007/978-3-031-19809-0\\_36](https://doi.org/10.1007/978-3-031-19809-0_36)
46. Wang, Z., et al.: Learning to prompt for continual learning. In: CVPR, pp. 139–149 (2022)
47. Wu, Y., et al.: Large scale incremental learning. In: CVPR, pp. 374–382 (2019)
48. Yan, S., Xie, J., He, X.: Der: dynamically expandable representation for class incremental learning. In: CVPR, pp. 3014–3023 (2021)
49. Yang, Y., Zhou, D.W., Zhan, D.C., Xiong, H., Jiang, Y., Yang, J.: Cost-effective incremental deep model: matching model capacity with the least sampling. IEEE Trans. Knowl. Data Eng. (2021)
50. Zeng, G., Chen, Y., Cui, B., Yu, S.: Continual learning of context-dependent processing in neural networks. Nat. Mach. Intell. **1**(8), 364–372 (2019)
51. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning, pp. 3987–3995. PMLR (2017)
52. Zhang, J., et al.: Class-incremental learning via deep model consolidation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1131–1140 (2020)
53. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: CVPR, pp. 13208–13217 (2020)



# Conditioned Prompt-Optimization for Continual Deepfake Detection

Francesco Laiti<sup>1</sup>, Benedetta Liberatori<sup>1(✉)</sup>, Thomas De Min<sup>1</sup>,  
and Elisa Ricci<sup>1,2</sup>

<sup>1</sup> University of Trento, Trento, Italy  
[benedetta.liberatori@unitn.it](mailto:benedetta.liberatori@unitn.it)

<sup>2</sup> Fondazione Bruno Kessler, Trento, Italy

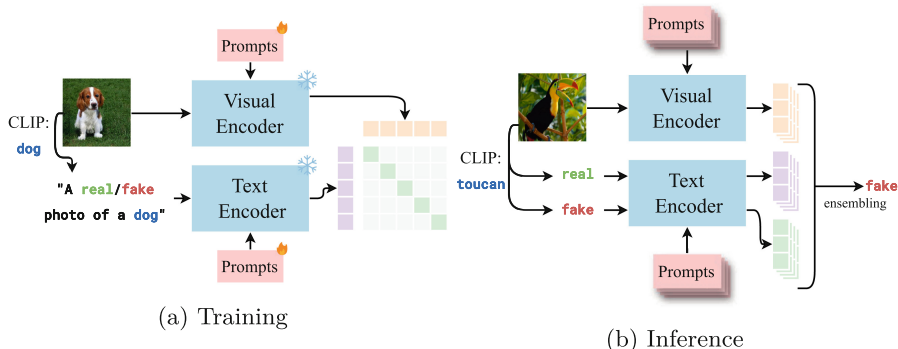
**Abstract.** The rapid advancement of generative models has significantly enhanced the realism and customization of digital content creation. The increasing power of these tools, coupled with their ease of access, fuels the creation of photorealistic fake content, termed deepfakes, that raises substantial concerns about their potential misuse. In response, there has been notable progress in developing detection mechanisms to identify content produced by these advanced systems. However, existing methods often struggle to adapt to the continuously evolving landscape of deepfake generation. This paper introduces Prompt2Guard, a novel solution for exemplar-free continual deepfake detection of images, that leverages Vision-Language Models (VLMs) and domain-specific multi-modal prompts. Compared to previous VLM-based approaches that are either bounded by prompt selection accuracy or necessitate multiple forward passes, we leverage a prediction ensembling technique with read-only prompts. Read-only prompts do not interact with VLMs internal representation, mitigating the need for multiple forward passes. Thus, we enhance efficiency and accuracy in detecting generated content. Additionally, our method exploits a text-prompt conditioning tailored to deepfake detection, which we demonstrate is beneficial in our setting. We evaluate Prompt2Guard on CDDDB-Hard, a continual deepfake detection benchmark composed of five deepfake detection datasets spanning multiple domains and generators, achieving a new state-of-the-art. Additionally, our results underscore the effectiveness of our approach in addressing the challenges posed by continual deepfake detection, paving the way for more robust and adaptable solutions in deepfake detection. Source code is available at <https://github.com/laitifranz/Prompt2Guard>.

**Keywords:** Deepfake detection · Incremental Learning · Prompt Learning · Multi-Modal Learning · Contrastive Learning

## 1 Introduction

The rapid evolution of generative artificial intelligence has revolutionized the digital content creation, enabling unprecedented levels of realism, customization, and accuracy [6, 19, 36, 38]. The ease of access to these technologies has

been crucial in driving their advancement, making powerful tools available to a broader audience beyond researchers, thereby removing barriers for non-expert users. This progress has led to photorealistic fake images and videos, *i.e.* *deep-fakes*, raising significant concerns regarding their potential for malicious use. With human discernment facing significant challenges in distinguishing between real and generated fake images [28], urgent attention is needed to develop effective detection mechanisms capable of accurately identifying content produced by these advanced systems.



**Fig. 1. Overview of the proposed method.** Prompt2Guard addresses the task of domain incremental deepfake detection. The training (a) is performed on a sequence of datasets, coming from different domains. At inference time (b) the model classifies the input image into real or fake, without domain knowledge.

Significant progress has been achieved on the deepfake detection as well, with state-of-the-art detectors capable of identifying images generated using GANs and diffusion models [5, 41]. However, these methods primarily function within a stationary scenario, wherein a large amount of relatively homogeneous deepfake content is presented at training time. This ideal scenario is often not reflective of the real-world landscape. In practice, likely heterogeneous deepfakes are continuously produced using novel and unseen architectures, presenting a constantly evolving landscape for detection methods to navigate. To effectively tackle this challenge, modern deepfake detectors must be able to adapt to the latest generators without succumbing to catastrophic forgetting. Maintaining the ability to detect content from diverse generators is crucial, as older generators continue to pose a significant threat.

While some continual deepfake detection benchmarks have been introduced lately [26, 54], the field still lacks comprehensive exploration, with few methods tackling deepfake detection in the incremental setting. Capitalizing on the generalization abilities of Vision-Language Models (VLMs), recent methods have shown promise in leveraging the encoded knowledge of these models for deepfake detection [30, 45].

These methods adapt VLMs to Domain Incremental Learning (DIL) by learning specific prompts for each task (generator) at training time, thereby maintaining independence in the training process. At test time, they either require inferring the generator to select the appropriate prompts [45] or, when undecided, performing multiple forward passes and aggregate information from different parameters [30]. As a result, they are either constrained by task selection accuracy or necessitate expensive multiple forwards to output a single prediction. Furthermore, while VLMs demonstrate potential in assessing the authenticity of visual content, their application in deepfake detection often oversimplifies the problem, treating the task as standard binary classification.

Inspired by these observations, we propose Prompt2Guard, a novel solution for exemplar-free continual deepfake detection that leverages VLMs and domain-specific multi-modal prompts, as illustrated in Fig. 1. Compared to previous VLM-based methods, our solution is specifically tailored for deepfakes and solves the task selection problem with a prediction ensembling that does not require multiple forward passes. We evaluate the proposed approach on the challenging CDDB benchmark [26], consisting of a sequence of datasets coming from different image generators, achieving state-of-the-art results.

Our contributions can be summarized as follows:

1. We present Prompt2Guard, a novel VLM-based exemplar-free DIL strategy that leverages multi-modal prompts. These prompts are read-only and do not alter the VLM internal representation. As a result, we can ensemble prediction scores from different tasks without requiring multiple forward passes, enhancing accuracy and efficiency.
2. Additionally, we propose a text-prompt conditioning procedure specifically tailored to deepfake detection and show its effectiveness.
3. We empirically show the capabilities of the proposed method, achieving state-of-the-art results in task-wise average accuracy without incurring catastrophic forgetting on the CDDB benchmark [26].

## 2 Related Work

**Deepfake Detection.** The field of media forensics has a long history of utilizing traditional tools to analyze synthetic images, including techniques such as identifying resampling artifacts [33], JPEG quantization [2], image splicing [22], and Photoshop warping [43]. With the democratization of synthetic image creation through deep generative methods, recent studies have focused on employing deep discriminative methods to detect such manipulated content, particularly for GAN-based approaches [11]. Rössler et al. [39] train an Xception [13] for detecting deepfake images of faces. Chai et al. [11] employ limited receptive fields to identify the most indicative patches, demonstrating that they contain sufficient cues for detecting images as real or fake. Wang et al. [44] show that CNN-generated images share common flaws and a ResNet-50 [20] with suitable data augmentations can generalize across generators.

**Vision-Language Models.** Vision-language models (VLMs), pioneered by CLIP [35], are pre-trained on a vast amount of web-crawled image-text pairs to learn joint visual-text embedding spaces. These models have demonstrated outstanding performance in various downstream tasks, especially in zero-shot image classification [23, 49]. While VLMs exhibit robust generalization capabilities, adapting them to specific tasks is challenging. Recent studies in VLMs involve prompt learning to adapt pre-trained models to downstream tasks using affordable-sized datasets. CoOp [53] uses continuous vector prompts, which are concatenated and processed with text tokens. CoCoOp [52] further extends CoOp by leveraging a lightweight neural network to generate prompts conditioned on the input image. These works keep the pre-trained weights frozen, yet the learnable prompts still affect the model’s hidden representation through the attention mechanism. To prevent this internal representation shift, RPO [25] proposes to use a masked attention mechanism, which limits prompts to only read information from the attention-based interactions of the pre-trained model.

**Continual Learning.** To tackle catastrophic forgetting, early continual learning approaches proposed regularization terms to constrain network parameters from forgetting old knowledge when updated with new information [3, 9, 12, 18, 24, 27, 50]. Despite reducing forgetting during sequential updates, these methods fail to retain satisfactory performances after multiple updates. By allowing the storage of part of the old data in memory buffers, rehearsal-based approaches [8, 10, 34, 37, 48] have shown superior performance over memory-free approaches. However, storing data in a replay buffer for future rehearsal poses privacy-related concerns as data may be leaked. To preserve privacy while maintaining performances similar to replay-based methods, parameter-isolation approaches exploit pre-trained models and tune just a small fraction of parameters for each update [16, 29, 42, 46, 47, 51], which are selected at inference time through a query-key selection mechanism. In particular, S-Prompts [45] tackles the domain-incremental learning problem (DIL) by tuning CLIP pre-training [35] on domain-independent sets of vision and language prompts. Since the domain shift across incremental steps is high in DIL, the query-key matching likely targets the best set of tuned prompts. MoP-CLIP [30], instead, proposes a mixture of prompt-tuned CLIP models to address the poor out-of-distribution performance of S-Prompts and DIL methods.

In this work, we propose to tune CLIP vision and language encoders with multi-modal read-only prompts [25], which allow us to compute multiple and parallel representations for each image, thus, computing the final prediction as a weighted sum of domain-specific predictions.

## 3 Preliminaries

### 3.1 Problem Formulation

In deepfake detection, the objective is to train a model capable of distinguishing real images from synthetically generated ones. Formally the model  $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ ,

parameterized by  $\theta$ , maps images from the input space  $\mathcal{X}$  to the binary semantic space  $\mathcal{Y} = \{0, 1\}$ , where generated samples should be predicted as 1. In this work, we tackle the problem of deepfake detection in an incremental learning scenario, a particular instance of domain incremental learning, where  $f_\theta$  must be trained sequentially over non-stationary datasets. Let  $\mathcal{D} = \{\mathcal{D}^1, \dots, \mathcal{D}^T\}$  be the sequence of datasets, the  $k$ -th dataset  $\mathcal{D}^k = \{(x_i, y_i)\}_{i=1}^{N_k}$  is composed of real and generated images with their corresponding semantic annotation. At each step, synthetic images are generated with a different generator  $\mathcal{G}^k$ , thus, the input data distribution shifts from task to task. Given two distinct tasks  $k$  and  $m$ , with  $k \neq m$ , the distributions of the two tasks are different, *i.e.*  $p(\mathcal{X}^k) \neq p(\mathcal{X}^m)$ . Given a new domain, DIL aims to improve the model’s performance on the latest distribution, while avoiding the loss of knowledge for past domains. At inference time, the model must classify the input image without knowing the domain. In the following, we will interchangeably use the terms *domain* and *task*.

### 3.2 Prompt Tuning

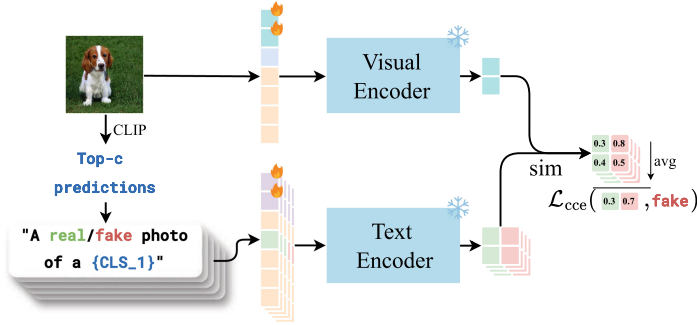
We follow previous works in the field [30, 45] and fine-tune CLIP [35] on the sequential datasets. Previous works in DIL [30, 45] exploit prompt tuning for adapting CLIP to the incremental detection of deepfakes. As prompts are specific for particular types of generated data, the training procedure of these approaches is independent for each generator. This reduces the risk of forgetting, as prompts, once trained, are kept frozen throughout the entire lifetime of the model, creating distinct subspaces for each domain’s knowledge rather than relying on a shared feature space for all tasks, thereby reducing the interference between old and new domains. However, training task-specific prompts forces previous methods to guess prompts to use at inference time, and this operation must be performed for each image. Images are then selected using the guessed prompts. This limits previous methods since wrong prompt selection results in lower model accuracy. MoP-CLIP [30] solves this issue by forwarding the target image multiple times with different trained prompts when the query-key selection mechanism has low confidence. However, this means MoP-CLIP has to forward the target image as many times as the number of tasks encountered by the model, which does not scale well in practice.

## 4 Prompt2Guard

The proposed method Prompt2Guard employs a pre-trained CLIP model as  $f_\theta$ , consisting of an image encoder  $\mathcal{E}_I$  and a text encoder  $\mathcal{E}_T$ , as shown in Fig. 2. This section details its main components: text-prompt conditioning, continual read-only prompts, and prediction ensembling.

### 4.1 Text-Prompt Conditioning

Contrary to previous DIL methods [30, 45], our objective is to design a tailored methodology for deepfake detection in an incremental setting. To this extent,



**Fig. 2. Illustration of the training.** The prepended prompts are the only learnable parameters (🔥), while the encoders are kept frozen (❄️).

in detecting synthetic data the model should focus more on specific attributes of the image (*i.e.* visual artifacts or inconsistencies) than on the image content. However, it has been observed that CLIP focuses on spurious and core features when classifying an image [4]. Thus, ignoring such spurious correlations can force CLIP to pinpoint salient artifacts in the image, making it more robust in detecting deepfakes. Similar to [4], we aim to focus CLIP attention on features that are more relevant for the synthetic content detection. We propose to infer the object’s class in a zero-shot fashion using CLIP and to use such information to condition the textual prompts, during training and inference. Nevertheless, the semantic space of image classes is usually unknown in deepfake detection datasets, thus we pre-define a set of classes  $\mathcal{C}$ . Given an input image  $x$ , we predict a category  $c^* \in \mathcal{C}$  as:

$$c^* = \arg \max_{c \in \mathcal{C}} \text{sim}(\mathcal{E}_I(x), \mathcal{E}_T(c)) \quad (1)$$

where  $\text{sim}(\cdot, \cdot)$  is the cosine similarity, computed as  $\text{sim}(u, v) = (u \cdot v) / \|u\| \|v\|$ . In this step, we augment the class names with the textual prompt “a photo of a {CLS}” [35]. Then, we use the predicted category to condition the textual prompt, obtaining as a result the prompt “a {real/fake} photo of a {c\*}”. In practice, instead of using just the highest-scoring one, we consider the first top- $c$  predicted classes and get  $c$  conditioned prompts. This is because the label set  $\mathcal{C}$  is agnostic to the categories in the dataset sequence  $\mathcal{D}$ . By considering the top- $c$  predictions, we account for potential uncertainties and provide more contextual information about the input image. For ease of reading, we will consider  $c = 1$  in the following. The obtained conditioned textual prompts are then used for both training and inference time, as shown in Fig. 1.

## 4.2 Continual Read-Only Prompts

To avoid solely relying on a single prompt while maintaining a low computational overhead, we propose to employ read-only prompts [25] as a substitute for



prompt-tuning. In particular, read-only prompts do not alter the internal representation of CLIP, and thus, at inference time we can concatenate prompts from different tasks and prepend them to the input. Let  $p_v^k \in \mathbb{R}^{L \times D_v}$  be the visual read-only prompt of task  $k$ , with length  $L$  and embedding dimension  $D_v$ , and let  $p_t^k \in \mathbb{R}^{L \times D_t}$  be the correspondent textual prompt for task  $k$ . Then, at task  $k$ , these are prepended to the visual and text encoder input as described in Sect. 3.2. The output of both encoders is dropped except for prepended prompts, which are the only trainable parameters in our setting. To train prompts, we employ a contrastive cross-entropy loss. Specifically, if we consider the case of a fake image, the loss is computed for each pair of output prompts as follows:

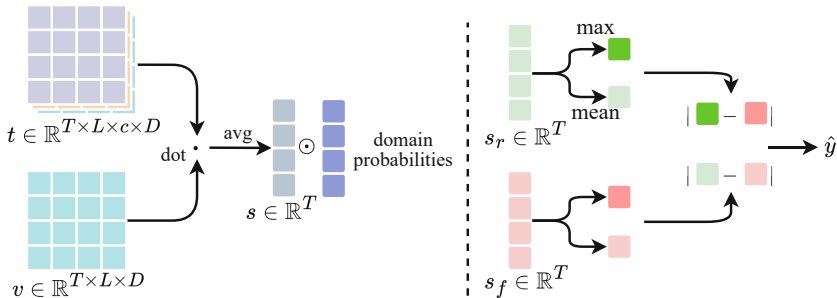
$$\mathcal{L}_{cce} = \frac{\exp(\overline{\text{sim}}(v^k, f^k))}{\exp(\overline{\text{sim}}(v^k, r^k)) + \exp(\overline{\text{sim}}(v^k, f^k))} \quad (2)$$

where  $v^k$ ,  $r^k$ , and  $f^k \in \mathbb{R}^{L \times D}$  are the visual, real, and fake output prompts, and  $\overline{\text{sim}}(\cdot, \cdot)$  is the average cosine similarity between text and visual prompts:

$$\overline{\text{sim}}(v, f) = \frac{1}{L} \sum_{i=1}^L \frac{v_i \cdot f_i}{\|v_i\| \|f_i\|} \quad (3)$$

Analogously, for a real image, the similarity at the numerator is computed between  $v^k$  and  $r^k$ , while the denominator is unaltered. As we mention in Sect. 4.1, instead of considering just the top class predicted by CLIP, we use the top- $c$  ones. As a result, we also average (3) across  $c$  classes aside from the length dimension (Fig. 3).

### 4.3 Predictions Ensembling



**Fig. 3. Illustration of the ensembling.** We compute the average similarity from the visual and textual prompts  $v$  and  $t$  obtained from the respective encoders. Then we weight the scores with the domain probabilities. This is repeated for real and fake textual prompts. The obtained  $s_r$  and  $s_f$  are used to obtain the predicted class  $\hat{y}$ .

As introduced in Sect. 4.2, at inference time our method does not require estimating the optimal set of parameters for each image compared to previous approaches. Instead, Prompt2Guard can leverage properties of read-only prompts to avoid altering the internal representation of CLIP. In practice, read-only prompts are unaware of other prompts in the forward pass, thus, prompts of different tasks do not influence their behavior. Therefore, this allows us to concatenate prompts of all seen domains, where each will focus on aspects of the image that are salient for a specific image generator. Formally, the input and output of the visual and text encoder are defined as follows:

$$\mathcal{E}_I([\{p_v^k\}_{k=1}^T, x_{cls}, x_{img}]) = \{v^k\}_{k=1}^T \quad (4)$$

$$\mathcal{E}_T([\{p_t^k\}_{k=1}^T, \text{a real photo of a CLS}]) = \{r^k\}_{k=1}^T \quad (5)$$

$$\mathcal{E}_T([\{p_t^k\}_{k=1}^T, \text{a fake photo of a CLS}]) = \{f^k\}_{k=1}^T \quad (6)$$

where (4) shows the input and output of the visual encoder, and (5) and (6) respectively the real and fake input and output prompts of the text encoder. To assign a score to a target image, we first use (3) to compute the average similarity for each pair of task prompts,  $(v^k, r^k)$  and  $(v^k, f^k)$ . As a result, we obtain two score vectors,  $s_r \in \mathbb{R}^T$  and  $s_f \in \mathbb{R}^T$ , that respectively represent the prompts confidence in predicting whether the image is real or generated. We scale each score vector entry by the likelihood that the generator corresponding to the entry has generated the image. In practice, we follow previous works [30, 45] and use a k-means classifier on the CLS token of the image to extract the probability distribution, and scale scores vector entries by the computed likelihoods. This allows for modulating the confidence of task prompts based on the likelihood that the image belongs to a specific domain. The task scores with the highest magnitude usually lead to better predictions, however, when confidence is low, exploiting the decisions of all task parameters leads to better accuracies (refer to Table 4). Thus, we compute both the maximum and mean of predictions:

$$s_r^* = \max\{s_r^1, \dots, s_r^T\}, \quad s_f^* = \max\{s_f^1, \dots, s_f^T\}, \quad (7)$$

$$\bar{s}_r = \frac{1}{T} \sum_{k=1}^T s_r^k, \quad \bar{s}_f = \frac{1}{T} \sum_{k=1}^T s_f^k, \quad (8)$$

Then, for the final prediction of the model, we use the score with the maximum confidence if the relative maximum confidence,  $|s_r^* - s_f^*|$  is greater than the relative mean confidence,  $|\bar{s}_r - \bar{s}_f|$ , otherwise, we uses the mean of logits:

$$\hat{y} = \begin{cases} \arg \max\{s_r^*, s_f^*\}, & \text{if } |s_r^* - s_f^*| \geq |\bar{s}_r - \bar{s}_f| \\ \arg \max\{\bar{s}_r, \bar{s}_f\}, & \text{otherwise} \end{cases} \quad (9)$$

By analyzing the confidence of the predictions, Prompt2Guard can automatically decide whether to use a mixture of experts or the score with the highest confidence, improving performance.

## 5 Experiments

**Dataset.** We perform experiments on the continual deepfake detection benchmark CDDB [26]. It gathers deepfakes from different generative models, gradually introduced to simulate the real-world deepfake’s evolution. In particular, it designs three different evaluation setups, *i.e.*, Easy, Hard, and Long. We select the most challenging, *i.e.*, the Hard sequence task (CDDB-Hard) in order to be comparable with previous methods. In particular, it consists of learning on 5 sequential deepfake detection domains, which are GauGAN [31], BigGAN [7], WildDeepfake [55], WhichFaceReal [1], and SAN [15] respectively.

**Metrics.** We perform the evaluation in terms of task-wise average accuracy (AA), which computes the average of all the task-based accuracies, as well as the average forgetting degree (AF), measuring the average decrease in accuracy on previous tasks after learning new tasks. In addition, we show the task-agnostic average accuracy (TAA), *i.e.* the accuracy of predictions calculated over all the images without considering the task, at the end of the training.

**Implementation Details.** We use CLIP (ViT-B/16), therefore  $D = 512$ . We set the length of both visual and textual read-only prompts as  $L = 7$  and use the top- $c$  classes with  $c = 5$ . For the closed-set of categories  $\mathcal{C}$  we use ImageNet-1k [40] classes for datasets containing images from general context and six hand-crafted ones for face datasets. The six face classes are obtained as a cross product between {young, middle-aged, old} and {male, female}. We use the SGD optimizer with a learning rate of 0.01 and cosine annealing with a constant warm-up of one epoch. We use 20 epochs per domain. Input images are resized to a resolution of  $224 \times 224$ , and the data augmentation consists of horizontal flipping, random cropping, and color jittering.

### 5.1 Comparative Results

We compare Prompt2Guard against several state-of-the-art methods including: non-prompting approaches such as EWC [24], LwF [27], LUCIR [21], iCaRL [37], and LRCIL [32] and prompting-based methods such as L2P [47], DyTox [17], S-Prompts [45], MoP-CLIP [30]. Our method is based on an exemplar-free DIL approach, thus we can assume as principal competitors EWC, LwF, DyTox, L2P, S-Prompts, and MoP-CLIP.

Table 1 presents the results on CDDB-Hard. The proposed Prompt2Guard outperforms previous methods, either exemplar-free or replay-based, delivering significantly better results in terms of AA. In particular, it surpasses the state-of-the-art S-Prompts by +1.63% on the AA and achieves a low AF of -0.71%. Figure 4 further illustrates the AA curve on CDDB-Hard of the proposed method, with and without the text-prompt conditioning, and of the competitor S-Prompts.

Table 2 shows the task-wise accuracy on each domain, the task-wise average accuracy (AA), and the task-agnostic average accuracy (TAA) of the proposed Prompt2Guard against the main competitor S-Prompts. The key insight is that our method achieves good accuracy across all the tasks, including the last

**Table 1. Results on CDDB-Hard.** Blue is **our method** and the best results are in **bold**. We also report if methods are prompt-based and their buffer size.

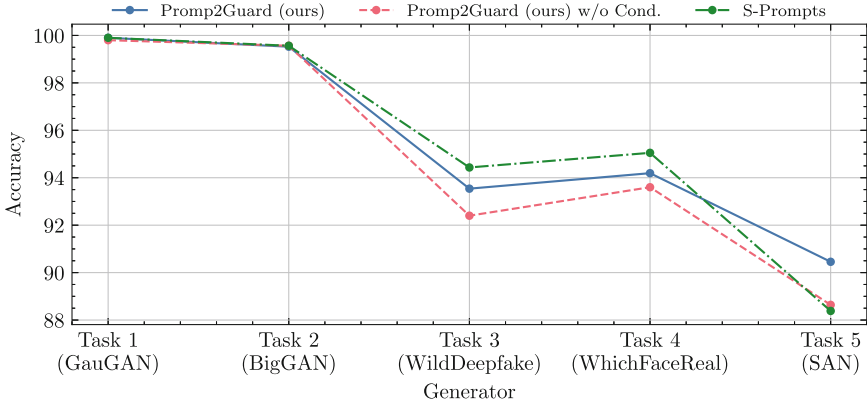
Method	Prompts	Buffer size	AA $\uparrow$	AF $\uparrow$
LRCIL [32]	×		76.39	-4.39
iCaRL [37]	×	100 samples/class	79.76	-8.73
LUCIR [21]	×		82.53	-5.34
LRCIL [32]	×		74.01	-8.62
iCaRL [37]	×	50 samples/class	73.98	-14.50
LUCIR [21]	×		80.77	-7.85
DyTox [17]	✓		86.21	-1.55
EWC [24]	×		50.59	-42.62
LwF [27]	×		60.94	-13.53
DyTox [17]	✓		51.27	-45.85
L2P [47]	✓	0 samples/class	61.28	-9.23
S-iPrompts [45]	✓		74.51	-1.30
MoP-CLIP [30]	✓		88.54	-0.79
S-liPrompts [45]	✓		88.65	<b>-0.69</b>
Prompt2Guard	✓		<b>90.28</b>	-0.71

and more challenging one, even if the accuracies in the previous tasks are slightly lower when compared to those of S-Prompts. In particular, S-Prompts obtains a task-wise accuracy of 68.89% on the last domain, while our Prompt2Guard gains a +12.22% improvement. Prompt2Guard benefits from the ensembling described in Section 4.3, particularly on samples from the last domain SAN, which corresponds to low domain classification accuracy, as shown in the confusion matrix in Fig. 5. Despite BigGAN having the lowest domain classification accuracy, its task-wise accuracy remains high. Therefore the model leverages the ensembling and correctly classifies images as real or fake, even when the domain is misclassified.

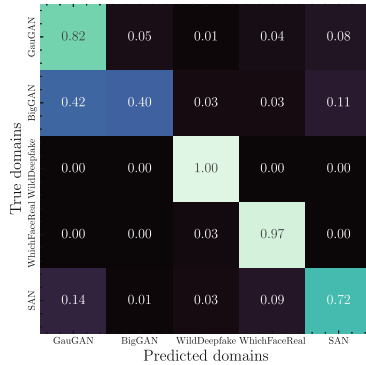
**Table 2. Comparison of task-wise accuracy across different domains, AA and TAA (%).** We show task-wise accuracy for each task of CDDB-Hard, both for S-Prompts and **our proposed method**.

Method	Dataset					Metrics	
	GauGAN	BigGAN	WildDeepfake	WhichFaceReal	SAN	AA $\uparrow$	TAA $\uparrow$
S-Prompts [45]	<b>99.30</b>	<b>96.75</b>	<b>82.06</b>	<b>96.25</b>	68.89	88.65	<b>91.54</b>
Prompt2Guard	98.70	94.38	81.73	95.50	<b>81.11</b>	<b>90.28</b>	90.98

Figure 6 presents the qualitative results of our proposed Prompt2Guard in detecting deepfakes from all the tasks in CDDB-Hard. We report the top- $c$  pre-



**Fig. 4.** Accuracy across tasks. We show the task-wise average accuracy (AA) values for Prompt2Guard and for the competitor S-Prompts across all the tasks of CDDB-Hard. We also show Prompt2Guard w/o conditioning, *i.e.* without the step described in Section 4.1. We plot the AA computed up to the  $i$ -th domain, against the domain index.

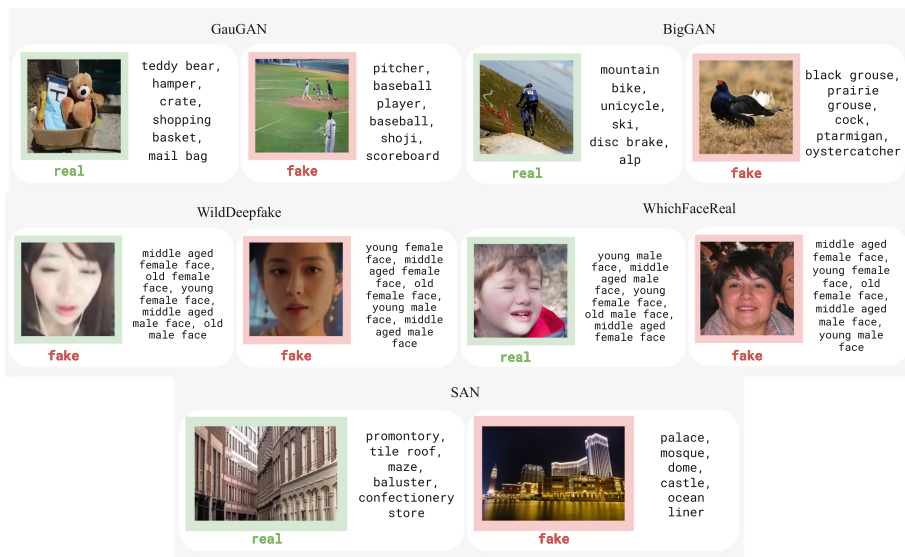


**Fig. 5. Task Confusion.** Confusion matrix of the domain classification of the proposed Prompt2Guard on CDDB-Hard.

dicted classes used for text-prompt conditioning, as described in Section 4.1. The model is capable of detecting deepfakes for most of the cases. Moreover, the predicted classes are consistent with the content of the images. We observe that using more than one class is beneficial, specifically when more objects are present.

## 5.2 Ablations

We ablate the proposed method Prompt2Guard on CDDB-Hard to validate the effectiveness of our design choices. First, we analyze the text-prompt condition-



**Fig. 6. Qualitative results.** We show the prediction of Prompt2Guard on test images from each task of CDDDB. The colored frame around the image indicates the ground truth class, while the text underneath is the predicted one. We report on the right the top- $c$  classes predicted and used for text-prompt conditioning.

ing described in Sect. 4.1, then the ensembling of the predictions detailed in Section 4.3.

**Text-Prompt Conditioning.** In Table 3 we assess the efficacy of conditioning the textual prompts on the category classified via zero-shot CLIP. When this step is added, we gain a +1.64% improvement in the AA and +0.14% in the AF. Our experiments validate the effectiveness of this choice, which lets the model focus more on salient artifacts relevant to deepfake detection rather than on the objects present in the image.

**Table 3. Ablation on text-prompt conditioning.** We report the results with and without conditioning the textual prompts on the category classified by zero-shot CLIP. Blue is our configuration .

Method	Text-prompt conditioning	AA $\uparrow$	AF $\uparrow$
Prompt2Guard	×	88.64	-0.85
Prompt2Guard	✓	<b>90.28</b>	<b>-0.71</b>

**Prediction Ensembling.** In Table 4 we compare three different ensembling techniques that can be used to obtain the final prediction  $\hat{y}$ . Directly averaging

the scores across the tasks (termed here as *mean*) produces the worst results. Using always the scores with the maximum confidence (*max*) results in better performance. Lastly, the highest results, both in AA and AF, are achieved with the ensembling defined in (9) (termed here as *max & mean*).

**Table 4. Ablation on prediction ensembling.** We compare different ensembling choices for the prediction. Blue is **our configuration**.

Method	Prediction ensembling	AA $\uparrow$	AF $\uparrow$
Prompt2Guard	mean	83.41	-1.47
Prompt2Guard	max	89.98	-1.15
Prompt2Guard	max & mean	<b>90.28</b>	<b>-0.71</b>

## 6 Conclusions

In this work we address the challenging problem of continual deepfake detection. We propose Prompt2Guard, a novel exemplar-free solution that leverages VLMs, read-only multi-modal prompts, and a text-prompt conditioning specifically tailored to the task. Our experimental evaluation confirms the effectiveness of Prompt2Guard in achieving state-of-the-art results in task-wise average accuracy on the challenging CDDDB benchmark. As future work, we plan to extend our method beyond the use of a closed label set, harnessing the power of vocabulary-free classification [14], and to evaluate it on images coming from more recent generators, *e.g.* including diffusion-based deepfakes. Additionally, we plan to address the scalability limitations of our method, which are constrained by the token length limitation of the VLM’s text encoder. This restriction limits the number of domains that learnable prompts can represent at inference time, thereby constraining the number of tasks that can be handled simultaneously in a DIL setting.

**Acknowledgments.** We acknowledge the CINECA award under the ISCRA initiative, for the availability of HPC resources. This work was also sponsored by PNRR FAIR - Future AI Research (PE00000013), funded by NextGeneration EU and supported by the EU project AI4TRUST (No.101070190). Thomas De Min is funded by NextGeneration EU.

## References

1. Which face is real?. <https://www.whichfaceisreal.com/>
2. Agarwal, S., Farid, H.: Photo forensics from jpeg dimples. In: WIFS (2017)
3. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: ECCV (2018)

4. An, B., Zhu, S., Panaitescu-Liess, M.A., Mummadi, C.K., Huang, F.: More context, less distraction: zero-shot visual classification by inferring and conditioning on contextual attributes. In: ICLR (2023)
5. Bird, J.J., Lotfi, A.: Cifake: image classification and explainable identification of ai-generated synthetic images. *IEEE Access* (2024)
6. Blattmann, A., et al.: Align your latents: high-resolution video synthesis with latent diffusion models. In: CVPR (2023)
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2019)
8. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. *NeurIPS* (2020)
9. Liu, Z., Freeman, W.T., Tenenbaum, J.B., Wu, J.: Physical primitive decomposition. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11216, pp. 3–20. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01258-8\\_1](https://doi.org/10.1007/978-3-030-01258-8_1)
10. Cha, H., Lee, J., Shin, J.: Co2l: contrastive continual learning. In: *ICCV* (2021)
11. Chai, L., Bau, D., Lim, S.-N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12371, pp. 103–120. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58574-7\\_7](https://doi.org/10.1007/978-3-030-58574-7_7)
12. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.S.: Riemannian walk for incremental learning: understanding forgetting and intransigence. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11215, pp. 556–572. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01252-6\\_33](https://doi.org/10.1007/978-3-030-01252-6_33)
13. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2016)
14. Conti, A., Fini, E., Mancini, M., Rota, P., Wang, Y., Ricci, E.: Vocabulary-free image classification (2023)
15. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: CVPR (2019)
16. De Min, T., Mancini, M., Alahari, K., Alameda-Pineda, X., Ricci, E.: On the effectiveness of layernorm tuning for continual learning in vision transformers. In: *ICCVW* (2023)
17. Douillard, A., Ramé, A., Couairon, G., Cord, M.: Dytox: transformers for continual learning with dynamic token expansion. In: CVPR (2022)
18. Fini, E., Da Costa, V.G.T., Alameda-Pineda, X., Ricci, E., Alahari, K., Mairal, J.: Self-supervised models are continual learners. In: CVPR (2022)
19. Ge, S., et al.: Preserve your own correlation: a noise prior for video diffusion models. In: CVPR (2023)
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. Huh, M., Liu, A., Owens, A., Efros, A.A.: Fighting fake news: image splice detection via learned self-consistency. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018*. LNCS, vol. 11215, pp. 106–124. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01252-6\\_7](https://doi.org/10.1007/978-3-030-01252-6_7)
23. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *ICML* (2021)
24. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *Proc. Nat. Acad. Sci.* (2017)



25. Lee, D., Song, S., Suh, J., Choi, J., Lee, S., Kim, H.J.: Read-only prompt optimization for vision-language few-shot learning. In: ICCV (2023)
26. Li, C., et al.: A continual deepfake detection benchmark: dataset, methods, and essentials. In: WACV (2023)
27. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* (2017)
28. Lu, Z., et al.: Seeing is not always believing: benchmarking human and model perception of ai-generated images. In: NeurIPS (2023)
29. McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E., van den Hengel, A.: Ranpac: Random projections and pre-trained models for continual learning (2024)
30. Nicolas, J., Chiaroni, F., Ziko, I., Ahmad, O., Desrosiers, C., Dolz, J.: Mop-clip: a mixture of prompt-tuned clip models for domain incremental learning. In: WACV (2024)
31. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019)
32. Pellegrini, L., Graffieti, G., Lomonaco, V., Maltoni, D.: Latent replay for real-time continual learning. In: IROS (2020)
33. Popescu, A., Farid, H.: Exposing digital forgeries by detecting traces of resampling. *IEEE Trans. Signal Process.* (2005)
34. Prabhu, A., Torr, P.H.S., Dokania, P.K.: GDumb: a simple approach that questions our progress in continual learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 524–540. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58536-5\\_31](https://doi.org/10.1007/978-3-030-58536-5_31)
35. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
36. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. *arXiv* (2022)
37. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: incremental classifier and representation learning. In: CVPR (2017)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
39. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: FaceForensics++: learning to detect manipulated facial images. In: ICCV (2019)
40. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. In: IJCV
41. Sha, Z., Li, Z., Yu, N., Zhang, Y.: De-fake: detection and attribution of fake images generated by text-to-image generation models. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (2023)
42. Smith, J.S., et al.: Coda-prompt: continual decomposed attention-based prompting for rehearsal-free continual learning. In: CVPR (2023)
43. Wang, S.Y., Wang, O., Owens, A., Zhang, R., Efros, A.A.: Detecting photoshopped faces by scripting photoshop. In: ICCV (2019)
44. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: CVPR (2020)
45. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: an occam’s razor for domain incremental learning. In: NeurIPS (2022)
46. Wang, Z., et al.: Dualprompt: complementary prompting for rehearsal-free continual learning. In: ECCV (2022)
47. Wang, Z., et al.: Learning to prompt for continual learning. In: CVPR (2022)
48. Wu, Y., et al.: Large scale incremental learning. In: CVPR (2019)
49. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: contrastive captioners are image-text foundation models. *TMLR* (2022)

50. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML (2017)
51. Zhou, D.W., Ye, H.J., Zhan, D.C., Liu, Z.: Revisiting class-incremental learning with pre-trained models: generalizability and adaptivity are all you need. arXiv (2023)
52. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022)
53. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. In: IJCV (2022)
54. Zhu, M., et al.: Genimage: a million-scale benchmark for detecting ai-generated image. In: NeurIPS (2023)
55. Zi, B., Chang, M., Chen, J., Ma, X., Jiang, Y.G.: Wildeepfake: a challenging real-world dataset for deepfake detection. In: ACM MM (2020)



# Plasticity Driven Knowledge Transfer for Continual Deep Reinforcement Learning in Financial Trading

Dimitrios Katsikas<sup>1</sup>(✉), Nikolaos Passalis<sup>1,2</sup>, and Anastasios Tefas<sup>1</sup>

<sup>1</sup> Department of Informatics, Faculty of Sciences, Computational Intelligence and Deep Learning (CIDL) Group, AIIA Lab, Thessaloniki, Greece  
{katsikasd,tefas}@csd.auth.gr, passalis@auth.gr

<sup>2</sup> Department of Chemical Engineering, Faculty of Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece

**Abstract.** The rapid growth in automated financial trading has highlighted the need for trustworthy agents capable of adapting to the dynamic and ever-changing nature of financial markets. From an algorithmic viewpoint, financial trading is essentially a complex, dynamic time series problem, characterized by unpredictable and noisy data. Deep Reinforcement Learning (DRL) has shown great promise in addressing this challenge. It naturally aligns with the objective of financial trading—maximizing rewards—without relying on unrealistic assumptions that do not hold true in such volatile and noisy time series data. However, the complexity of the problem still presents challenges for conventional DRL algorithms. To overcome these, the implementation of continual learning agents is crucial for their ability to adjust to changing market conditions. Our approach not only adapts continual learning techniques to dynamic time series but also introduces a novel knowledge transfer loss, which enhances the adaptation of our model. In our extensive evaluation, we show that this approach successfully balances the trade-off between maintaining knowledge of past patterns and adapting to new ones, enhancing the model’s trustworthiness and effectiveness in real-world time series problems, like financial trading.

**Keywords:** Continual learning · Deep reinforcement learning · Financial trading

## 1 Introduction

In recent years, the financial landscape has experienced a significant surge in automated trading, where algorithms and computational models are used to execute trades at a speed, scale and accuracy unattainable by human traders. This growth arises a concomitant issue, as similar trading strategies are massively adopted by numerous trading firms. This issue is known as “alpha decay” [19], where alpha refers to the measure of the excess return of an investment strategy

compared to a benchmark, such as a market index. Initially, a novel trading agent may generate substantial alpha, but as it becomes commonplace, the returns tend to diminish. This decay occurs because the market adapts and the edge that the agent provided vanishes. This phenomenon ties in with the Efficient Market Hypothesis (EMH), which asserts that asset prices reflect all available information at any given time [3]. Essentially, EMH posits that it is impossible to consistently achieve risk-adjusted returns that exceed the market average, as new information is rapidly incorporated into asset prices. In such a context, financial trading can be seen as a highly dynamic time series, highlighting the need for adaptive continual learning approaches in automated trading.

The recent breakthrough in the field of Deep Learning (DL) [21], has led to the creation of more advanced and sophisticated trading agents [23, 24, 27]. These agents leverage the enormous volumes of data amassed from financial markets, along with supplementary data from news articles and social media [15, 16] in order to make more accurate and profitable predictions. One particular area of DL that has seen remarkable advancements is Deep Reinforcement Learning (DRL) [7, 10, 11, 13] providing potent models that are trained on maximizing profits directly through their reward functions [2, 25], instead of approximating the task through handcrafted proxy problems. Indeed, traditional DL for trading typically relies on supervised learning [24], in which agents are trained using handcrafted labels. These labels attempt to take into account real trading conditions, such as commissions and other costs. However, they often fall short of mirroring real-world profits due to the multitude of factors at play. For instance, market volatility and the confidence level of the agent can greatly impact the agent’s performance, leading to significant discrepancies between predicted and actual profits or losses.

On the other hand, DRL presents a different approach [2, 25], by naturally enabling the incorporation of trading profits into the model’s rewards, which can be achieved alongside other market costs. This is possible through the use of simulated trading environments for training DRL agents. This approach bypasses the challenges of handcrafting complex labels and permits the trading agents to discern the positions worth taking, with predictable results based on the rewards received. As such, DRL agents are capable of directly fine-tuning the metrics that are critical to the task at hand, specifically the profits earned, within these simulated trading environments.

While DRL’s capability to accurately model profits within its training environment might initially appear sufficient for developing profitable trading agents, a major challenge exists. Most DRL methods are plagued by instability, resulting in agents with highly variable trading behavior, which do not perform robust in different market conditions. This inconsistency across different training runs or time periods substantially undermines the trustworthiness and reliability of the profits obtained, and consequently diminishes our confidence in these agents. To counter these challenges, it is imperative to adopt tailored training processes that bolster training stability when utilizing DRL techniques in financial trading applications [2, 25]. Furthermore, the discussed nature of financial markets

requires a method that goes beyond deploying static DRL trading agents, since the markets often evolve so rapidly that by the time these agents are put into use, they are already outdated and ill-suited to past market conditions.

In this work, we present a novel continual learning methodology for DRL agents, that involves implementing periodic updates to the agent in an effort to strike a balance between stability and plasticity. The stability-plasticity dilemma refers to the challenge of allowing a model to adapt to new data (plasticity) while retaining the knowledge it has previously acquired (stability) [12]. Balancing these two aspects is crucial for the model to perform effectively in dynamic environments like financial markets, where the distribution of data changes over time. The proposed method is built upon two key ideas. Firstly, it employs a dynamic experience replay mechanism with temporal focus to the current market conditions. This approach is designed to allow the model to not only adapt to the latest market conditions but also to retain previously learned patterns. Secondly, it involves utilizing a plasticity driven knowledge transfer process across multiple layers of our DRL neural network architectures [5]. This process includes forming for each update of our agent an ensemble of teacher models, each of them highly adept in different market conditions, aiming at improving the adaptation to new patterns. We performed an extensive experimental evaluation using a continual learning set-up for cryptocurrency trading, which forms a challenging dynamic time series environment, demonstrating solid improvements against both baseline static agents and greedily finetuned models, that are prone to catastrophic forgetting of previous knowledge.

The rest of the paper is structured as follows. In Sect. 2 we present a quick overview of related work in the field. Then, in Sect. 3, the proposed method is introduced and analytically derived, breaking down the motivation behind our methodology. The experimental setup and the experimental evaluation, are provided in Sect. 4. Finally, Sect. 5 concludes the paper.

## 2 Related Work

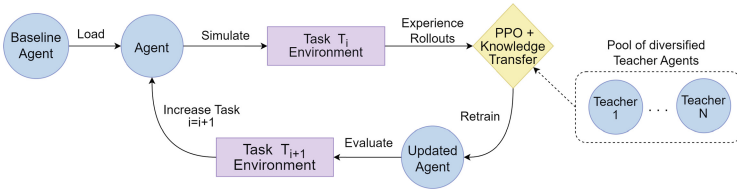
In financial trading, the use of direct price forecasts or reliance on expert-generated labels for training supervised models is considered sub-optimal [14]. Deep Reinforcement Learning (DRL) is argued to be more promising, as it optimizes performance directly by accounting for transaction costs, which can significantly differ between labels used in supervised learning and actual model predictions. Works such as [2, 27] have demonstrated the effectiveness of DRL in financial trading by employing profit or return-related metrics as the agent’s objectives. To further enhance agent performance, auxiliary DRL objectives and regularizers have been introduced [25], along with the integration of sentiment information, creating multimodal approaches for improved robustness [1]. Despite these significant contributions, the unique challenges of financial trading, characterized by its intricate complexities, dynamic environments, and high-frequency data streams, necessitate the development of specialized continual learning methodologies. To the best of our knowledge, this work is the first

to formulate and address the problem of continual adaptation of a DRL agent in a financial market, while tackling the issue of catastrophic forgetting of previous knowledge.

In the context of Continual Reinforcement Learning, approaches can be categorized into three clusters: explicit knowledge retention, leveraging shared structure, and meta-learning [8]. Explicit knowledge retention involves strategies such as parameter storage [9], knowledge distillation [20], and experience replay [6]. Leveraging shared structure focuses on exploiting task commonalities to streamline learning and enhance generalization across different tasks [26]. Lastly, meta-learning, with significant contributions like Model Agnostic Meta-Learning (MAML) [4], has played a crucial role in addressing multi-task and continual learning scenarios. Building upon these approaches, our study aims to develop a continual learning methodology for dynamic, time-dependent environments, typical in financial trading.

### 3 Proposed Method

In this section, we formulate the proposed methodology for financial trading through Deep Continual Reinforcement Learning. Initially, we outline the underlying Continual Reinforcement Learning framework. To tackle the inherent challenges, we introduce two novel approaches: Temporal Focused Sampling (TFS) experience replay, as well as plasticity driven knowledge transfer for controlling the knowledge retention and adaptation capabilities of the model.



**Fig. 1.** The proposed continual Reinforcement Learning framework consists of repetitively updating the employed agent on the current task and evaluating in on the next task. The proposed knowledge transfer interface is also shown in the diagram, which contains a dynamic pool of teacher models.

#### 3.1 Continual Learning Approach

In this work, we primarily focus on learning Deep Reinforcement Learning (DRL) policies through Policy Gradient-based methods. The Proximal Policy Optimization (PPO) serves as our fundamental basis, as it has achieved state-of-the-art results in numerous DRL problems [22]. However, this choice is not restrictive; the proposed method can be adapted to any other DRL approach with minimal

modifications. Our DRL agent interacts with a trading environment, utilizing historical price data as the state space and offering three trading actions: buy, sell, and hold, as the action space. To gather the necessary trajectories for training the DRL agents, in line with our objectives, we employ a rollout buffer. This mechanism generates episode rollouts within a specified range, capturing crucial data from each rollout, including rewards, predicted state values, and action probabilities. During optimization phases, the data stored in the experience replay memory is used to train the neural networks that form the agent’s actor and critic components.

Considering the dynamic nature of market price time series and their constantly shifting distribution, we define each trading period as a distinct task  $T$ , with its corresponding distribution  $D(T)$ . More specifically, we start with an initial task,  $T_{init}$ , followed by a sequence of non-overlapping tasks  $T_1, T_2, \dots, T_n$ , each occurring at a fixed frequency  $F$ . The initial task,  $T_{init}$ , is unique due to its use of a large volume of past data for training a baseline PPO agent, thereby establishing core knowledge. This agent forms the starting point in our continual learning framework, as illustrated in 1. For each subsequent task  $T_i$ , we apply our continual learning methodology to update the agent. A straightforward approach would involve collecting state transitions  $s, a, r, s'$  from  $D(T_i)$  for the replay buffer, and using them to finetune the static model. The updated policy is then employed in the next task  $T_{i+1}$ , with its performance evaluated in terms of Profit and Loss (PnL) for that sub-period. This process is repeated for all tasks, representing the sub-periods of the entire testing period. We name this simple approach as finetuning and it consists our continual learning baseline. However, finetuning, focuses solely on optimizing the current task without preserving performance on previous tasks, leading to catastrophic forgetting of old tasks. It is also highly sensitive to the update frequency  $F$ , often resulting in unstable and unreliable performance. Therefore, it is crucial to explore more sophisticated approaches that effectively mitigate catastrophic forgetting while adapting to new data distributions.

### 3.2 Temporal Focused Sampling Experience Replay

To address catastrophic forgetting, we propose a novel experience replay approach called Temporal Focused Sampling (TFS) experience replay. Experience replay methods are employed to help against catastrophic forgetting by maintaining a memory buffer of past experiences used alongside new data during retraining [6]. However, such an approach, does not take into consideration the temporal nature of dynamic time series. In TFS, the sampling mechanism of the replay buffer consists of a left tail Gaussian Distribution  $N(i, \sigma_\tau)$ , over the entire timeline of encountered tasks, centered at the currently considered task, symbolized by the index  $i$ , and with standard deviation  $\sigma_\tau$ , where  $\tau = \frac{n_{cur}}{n_{tot}}$  is a normalized time factor, defined as the ratio of the current epoch to the total number of epochs. In TFS,  $\sigma_\tau$  is not static; it evolves dynamically throughout the re-training process and its value at any given epoch is computed as:

$$\sigma_\tau = \sigma_{factor, \tau} \cdot \sigma_{max} + \sigma_{min}, \quad (1)$$

where  $\sigma_{\max}$  indicates the maximum standard deviation and plays an important role in controlling the Gaussian distribution’s spread across the complete task timeline. Conversely,  $\sigma_{\min}$  signifies the minimum standard deviation, resulting in narrower coverage around the samples of the current task  $T_i$ . The term  $\sigma_{\text{factor},\tau}$  is a scalar quantity fluctuating between 1 and 0, and it decays during the model’s iterative update process. It is responsible for modulating the influence of  $\sigma_{\max}$  on the standard deviation.

We discovered that using a sigmoid decay function, represented by  $\delta$ , effectively modulates  $\sigma_{\text{factor},\tau}$  throughout each update. The decay function  $\delta$  is defined as follows:

$$\delta(\tau, \lambda) = \frac{1}{1 + e^{-\lambda \cdot \tau}}, \quad (2)$$

where  $\lambda$  is a parameter that controls the steepness of the decay in the sigmoid function. Incorporating this sigmoid decay function into the model results in the update of  $\sigma_{\text{factor},\tau}$  at each epoch as follows:

$$\sigma_{\text{factor},\tau_{\text{next}}} = \sigma_{\text{factor},\tau} \cdot \delta(\tau, \lambda), \quad (3)$$

This updating scheme leads to a gradual decrease in  $\sigma_{\text{factor}}$ , which in turn assures a progressive reduction in the Gaussian distribution’s spread as the model retrains across epochs. This adaptive methodology allows for a temporal focused coverage of the task samples.

### 3.3 Knowledge Transfer Methodology

During our experiments, a notable phenomenon was observed. Agents, trained exclusively using TFS experience replay with their Gaussian distributions centered at various past time points, exhibited highly variable performance across different test periods. These models, which we have categorized into the Plasticity pool, symbolized as  $\mathcal{P}$ , demonstrated high performance during specific periods, but also significant volatility and losses during others, due to the dynamic nature of the price patterns.

To take advantage of that observation, after each task  $T_i$ , we select the top-performing models from the Plasticity pool in  $T_i$ , transferring their knowledge to our student agent, which is going to be updated and then employed in  $T_{i+1}$ . Relying on a single teacher from the Plasticity pool proved to be sub-optimal, due to the high volatility of these models. To address this, an ensemble  $\mathcal{E}_{T_i}$  of teachers is formed by selecting the top  $N$  performing models based on their performance in task  $T_i$ .

$$\mathcal{E}_{T_i} = \{t \in \mathcal{P} \mid t \text{ is among the top } N \text{ models in } T_i\}$$

The weights assigned to these teachers in the ensemble are computed by normalizing the PnL of these models in task  $T_i$ . For a teacher model  $t$  in  $\mathcal{E}_{T_i}$ , the weight is given by:

$$w_t^{T_i} = \frac{PnL_t^{T_i}}{\sum_{t' \in \mathcal{E}_{T_i}} PnL_{t'}^{T_i}}$$



As a distillation loss, we utilize a combination of two knowledge transfer techniques, shown to work . Firstly, the well-known soft label-based neural network distillation at the output layer of our architecture, as introduced in [5], as well as Probabilistic Knowledge Transfer (PKT) [17, 18] in the intermediate layers, which aims to align the internal representations of the teacher and student models in the feature space by minimizing the divergence between their corresponding probability distributions. We present a brief explanation of the two losses.

Considering the output soft label neural network distillation, let  $\pi^{(T)}(\alpha|\mathbf{s})$  and  $\pi^{(S)}(\alpha|\mathbf{s})$  represent the action probability distributions output by the teacher and student models respectively. Furthermore, let  $y^{(T)}(\alpha|\mathbf{s})$  and  $y^{(S)}(\alpha|\mathbf{s})$  denote the logits, or raw outputs, of the teacher and student models respectively. The soft labels, generated using the teacher model, produce a softer version of the probability distribution over actions, and is computed as follows:

$$q(\alpha|\mathbf{s}) = \frac{\exp\left(\frac{y^{(T)}(\alpha|\mathbf{s})}{T}\right)}{\sum_a \exp\left(\frac{y^{(T)}(a|\mathbf{s})}{T}\right)}, \quad (4)$$

where  $T$  is the temperature parameter, controlling the softness of the distribution. Similarly, a soft version of the student model’s output is calculated as:

$$p(\alpha|\mathbf{s}) = \frac{\exp\left(\frac{y^{(S)}(\alpha|\mathbf{s})}{T}\right)}{\sum_a \exp\left(\frac{y^{(S)}(a|\mathbf{s})}{T}\right)}. \quad (5)$$

The loss function optimized during knowledge transfer, which seeks to minimize the divergence between the soft distributions, is defined as:

$$L_D = -\frac{1}{N} \sum_{\mathbf{s} \in \mathcal{S}} \sum_{i=0}^{N_a} q(\alpha_i|\mathbf{s}) \log(p(\alpha_i|\mathbf{s})), \quad (6)$$

where  $\mathcal{S}$  is a set of  $N$  states sampled from the experience replay memory,  $N_a$  represents the number of available actions, and  $\alpha_i$  is the  $i$ -th available action. This loss function guides the student model to approximate the softened action probabilities of the teacher model, effectively transferring the knowledge encoded in the teacher’s probability distributions.

Regarding the PKT, let us denote the internal representations of the teacher model as  $f^{(T)}(\mathbf{s}) \in \mathbb{R}^M$ .  $f^{(T)}(\cdot)$  denotes the teacher model (up to the point where the representation is extracted)  $M$  is the dimensionality of the extracted representation from the teacher model and that of the student model as  $f^{(S)}(\mathbf{s}) \in \mathbb{R}^{M'}$ , where  $f^{(S)}(\cdot)$  denotes the student model (up to the point where the representation is extracted) and  $M'$  is the dimensionality of the extracted representation from the student model. To simplify the presentation of the proposed method, we defined as  $\mathbf{x}_i$  the internal representation of the teacher model when presented with the  $i$ -th state sampled from the buffer, i.e.,  $\mathbf{x}_i^{(T)} = f^{(T)}(\mathbf{s}_i)$ . The representation of the student model is similarly defined as  $\mathbf{x}_i^{(S)} = f^{(S)}(\mathbf{s}_i)$ . Then, we

employ kernel density estimation to estimate the conditional probability distributions of both the teacher and the student representations. Specifically, the conditional probability distribution for the teacher model is computed as:

$$p_{i|j}^{(T)} = \frac{K(\mathbf{x}_i^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma^2)}{\sum_{k=1, k \neq j}^N K(\mathbf{x}_k^{(T)}, \mathbf{x}_j^{(T)}; 2\sigma^2)}, \quad (7)$$

and for the student model as:

$$p_{i|j}^{(S)} = \frac{K(\mathbf{x}_i^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma^2)}{\sum_{k=1, k \neq j}^N K(\mathbf{x}_k^{(S)}, \mathbf{x}_j^{(S)}; 2\sigma^2)}. \quad (8)$$

Here,  $K(\mathbf{x}, \mathbf{y}; \sigma_2)$  denotes a symmetric kernel function with bandwidth  $\sigma$ . In this study, we utilize a kernel function that is based on the cosine similarity metric, following the observations reported in [18], which is defined as:

$$K_{\text{cosine}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \left( \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} + 1 \right), \quad (9)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  two vectors. This kernel function essentially measures the cosine of the angle between two vectors, which is scaled to the range  $[0, 1]$ . To quantify the divergence between the probability distributions of the teacher and student models, PKT makes use of the Kullback-Leibler (KL) divergence. The KL divergence is given by:

$$L_R = \sum_{i,j} p_{i|j}^{(T)} \log \left( \frac{p_{i|j}^{(T)}}{p_{i|j}^{(S)}} \right). \quad (10)$$

This KL divergence, referred to as the representation distillation loss, makes the student model’s representations more aligned with those of the teacher model.

Therefore, the final loss function of our method is formulated by combining the DRL loss  $L_{RL}$ , the distillation-associated loss  $L_D$ , and the representation knowledge transfer loss  $L_R$ :

$$L = L_{RL} + \alpha L_D + \beta L_R, \quad (11)$$

with  $\alpha$  and  $\beta$  serving as hyperparameters controlling the weight of the action distillation loss and representation-level distillation loss respectively, in the total loss.

## 4 Experimental Evaluation

### 4.1 Dataset and Feature Extraction

The dataset used in the conducted experiments consists of Cryptocurrency trading data for 15 Cryptocurrency/USDT pairs, such as BTC/USDT, ETH/USDT, LTC/USDT. We employ a sub-sampling technique based on High-Low-Close

(HLC) price levels to process this data and adapted 1-hour price intervals as the simulation stepping interval. We chose not to incorporate extra features because these basic features encapsulate the bulk of the market information and utilizing them enhances computational efficiency.

The agents receive an observation consisting of a window of preprocessed past price candles that have been normalized to ensure compatibility across different cryptocurrency pairs. The preprocessing produces the following features for each time step  $t$ :

$$\begin{aligned} - f_1(t) &= \frac{p_c(t)}{p_c(t-1)} - 1 \\ - f_2(t) &= \frac{p_h(t)}{p_h(t-1)} - 1 \\ - f_3(t) &= \frac{p_l(t)}{p_l(t-1)} - 1 \end{aligned}$$

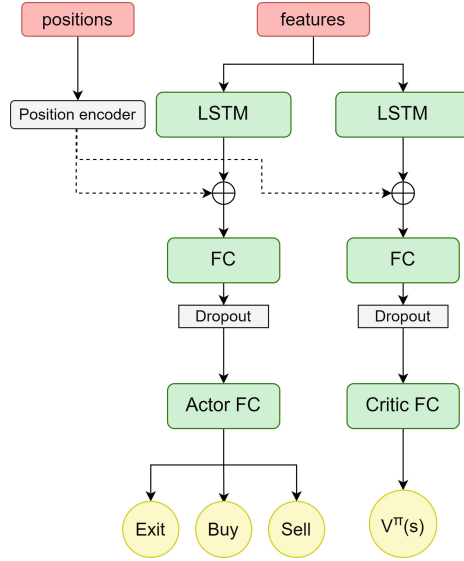
where  $p_h(t)$ ,  $p_l(t)$ , and  $p_c(t)$  represent the high, low, and close prices at time  $t$ , respectively. These features are percentage distances between sampled price values and encapsulate the range and variations within and across price candles. The observations also include the agent’s current market position, symbolized by a one-hot vector,  $\mathbf{x}_p(t)$ , of size 3, where  $[1, 0, 0]$  indicates no position,  $[0, 1, 0]$  denotes an active long position, and  $[0, 0, 1]$  signifies an active short position.

Data spanning from 1st January 2018 to 31st January 2022 constitute the past period, where the initial static baseline trading agent was trained on, while testing period is selected from 1st February 2021 to 27 January 2022, as this duration exhibits near-zero mean characteristics. The agent’s reward is the profit and loss (PnL) from its current position, adjusted for commission fees on action changes. PnL is calculated using the asset’s price time series and the agent’s position, with a fixed lot size. This ensures that PnL is based on percentage changes of the initial investment and not influenced by past profits or losses.

For the continual learning configuration, we experimented with four distinct update frequencies  $F$  - every 5, 10, 15, and 20 d, segmenting our 360-day testing period into 72, 36, 24, and 18 sub-periods or tasks, respectively. We compare the models based on both the cumulative PnL over the entire testing period (test PnL) and the average PnL of the updated models for each frequency, when backtested on the past period (past PnL), to examine knowledge retention.

## 4.2 Model Architecture

In our experiments, all trading agents utilize a uniform neural network architecture and hyperparameters to ensure an equitable comparison of the methodologies, which is shown in Fig. 2. The chosen architecture is composed of a Long Short-Term Memory (LSTM) layer with 32 hidden units, succeeded by one fully connected layer of 32 hidden units, employing Sigmoid Linear Unit (SiLU) as activation functions and dropout. These layers process the input time series, generating a representation for each time step. Subsequently, this representation is channeled into two distinct branches consisting of fully connected layers. The first branch serves as the actor and outputs a probability distribution across



**Fig. 2.** The employed trading agent’s Actor-Critic architecture comprises a shared base with a 32-unit LSTM layer and a 32-unit fully connected layer. The actor head generates action policy probabilities, while the critic head estimates state values.

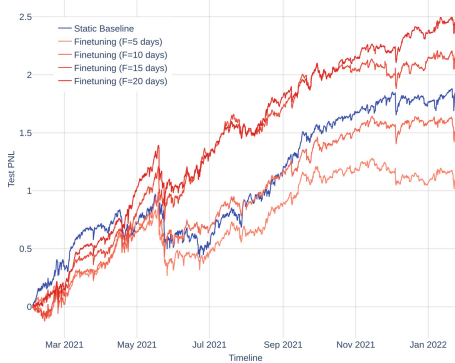
three potential actions: exit, buy, and sell. In contrast, the second branch, which constitutes the critic, evaluates the current state’s value based on the hidden representation.

Regarding hyperparameters, the agents were trained with RAdam optimizer with a learning rate set at  $5 \times 10^{-4}$ , a batch size of 32, dropout of 0.2, and a typical commission fee of  $2 \times 10^{-4}$  per trade. The initial baseline agent was trained for 300 epochs, while each update consists of 100 epochs of retraining. Each experiment was executed with five distinct seeds, and the results are presented as the average of those runs.

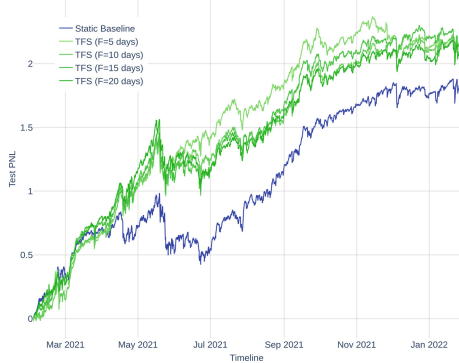
### 4.3 Results

In finetuning, the agent is sequentially updated for 100 epochs on the most recent task and subsequently evaluated on the next task. The experimental evaluation of finetuning is provided in Fig. 3, where the cumulative PnL over time for varying update frequencies is provided. When using a higher frequency, such as 5 or 10 d, the finetuned agent underperforms relative to the static baseline agent. However, the potential of continual learning is demonstrated when an update frequency of 15 and 20 d is used, surpassing the baseline. Despite this, finetuning clearly suffers by catastrophic forgetting, evidenced in Table 1.

In implementing the proposed TFS experience replay approach, we use a generic hyperparameter configuration for the sigma decay, to avoid overfitting the method to particular update frequencies. Let’s assume that for a given



**Fig. 3.** Finetuning against the static baseline agent.



**Fig. 4.** Finetuning with TFS experience replay against the static baseline agent.

update frequency measured in days  $F$ , each task comprises  $N_s$  samples, with  $N_s = F \times 24$ , given that we employ a 1-hour timeframe. We define  $\sigma_{\max}$  as 100 times the number of samples corresponding to the current task, as an approximation to a uniform distribution across the entire timeline. Concurrently,  $\sigma_{\min}$  is set equal to  $N_s/2$ , which allows for drawing approximately 95% of the samples from current task’s distribution  $D(T_i)$  in the terminal epochs of our update.

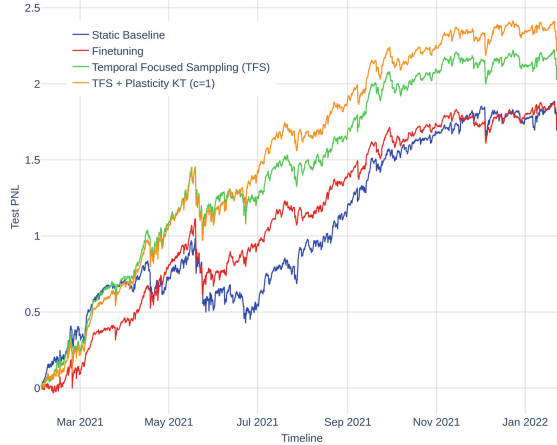
Using TFS experience replay while finetuning the agent, results in a higher and much more stable average test PnL across different  $F$  values, as shown in Fig. 4, while it manages to retain considerably improved performance in the past PnL as presented in Fig. 1. TFS mechanism provides an adaptive selection of experiences, offering a balance between past knowledge retention and adaptation to current patterns, as shown in these experiments.

**Table 1.** PnL Performance averaged over all update frequencies.

Models	Test PnL	Past PnL
Static Baseline	$1.83 \pm 0.29$	$5.95 \pm 0.25$
Finetuning	$1.79 \pm 0.57$	$3.12 \pm 0.16$
Temporal Focused Sampling (TFS)	$2.12 \pm 0.21$	<b><math>5.06 \pm 0.23</math></b>
TFS + Plasticity focused KT	<b><math>2.31 \pm 0.23</math></b>	$4.22 \pm 0.35$

After formulating a robust way to ensure model’s stability through TFS experience replay, we shift our focus on enhancing its plasticity to new market trends with knowledge transfer. Including the plasticity driven knowledge transfer loss in the retraining process, resulted in the highest average test PnL across all update frequencies  $F$ , as can be seen in Fig. 5. The ensemble of the most relevant teachers, improves student’s relevance to current market conditions. Despite coming with a significant drop in the past PnL, due to to the

plasticity-stability trade off, our proposed TFS + Plasticity focused KT method still gives some attention to old knowledge, which leads to notably higher past PnL than finetuning, as depicted in Table 1.



**Fig. 5.** Ablation study evaluating the impact of proposed components on cumulative test PnL during back-testing. Incremental enhancements to finetuning achieved with the introduction of TFS experience replay and Plasticity focused KT. Each line is averaged over all 4 different update frequencies.

## 5 Conclusions

In this paper, we presented a novel Deep Continual Reinforcement Learning methodology tailored for dynamic time series. The proposed method leverages TFS experience replay and employs ensemble teacher selection for plasticity driven knowledge transfer. The TFS experience replay ensures that the model remains consistent in historical data while adapting to new trends. The teacher selection, on the other hand, increases the adaptability of our agent, to patterns that are highly relevant to the current conditions. The obtained empirical results demonstrate that the proposed method achieves robust performance across different update frequencies. Future research could explore how to enable an agent to dynamically trigger such an update mechanism on its own, rather than relying on a predefined periodic update scheme as used in this study, with the aim of achieving a fully autonomous adaptive agent.

**Acknowledgements.** The research project “Energy Efficient and Trustworthy Deep Learning - DeepLET” is implemented in the framework of H.F.R.I call “Basic research Financing (Horizontal support of all Sciences)” under the National Recovery and Resilience Plan “Greece 2.0” funded by the European Union -NextGenerationEU (H.F.R.I. Project Number: 016762).

## References






1. Avramelou, L., Nousi, P., Passalis, N., Tefas, A.: Deep reinforcement learning for financial trading using multi-modal features. *Expert Syst. Appl.* **238**, 121849 (2024). <https://doi.org/10.1016/j.eswa.2023.121849>, <https://www.sciencedirect.com/science/article/pii/S0957417423023515>
2. Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q.: Deep direct reinforcement learning for financial signal representation and trading. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(3), 653–664 (2017). <https://doi.org/10.1109/TNNLS.2016.2522401>
3. Fama, E.: Efficient capital markets: a review of theory and empirical work. *J. Finance* **25**, 383–417 (1970)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: *Proceedings of the International Conference on Machine Learning*, pp. 1126–1135 (2017)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)
6. Isele, D., Cosgun, A.: Selective experience replay for lifelong learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
7. Keneshloo, Y., Shi, T., Ramakrishnan, N., Reddy, C.K.: Deep reinforcement learning for sequence-to-sequence models. *IEEE Trans. Neural Netw. Learn. Syst.* **31**(7), 2469–2489 (2019)
8. Khetarpal, K., Riemer, M., Rish, I., Precup, D.: Towards continual reinforcement learning: a review and perspectives (2022)
9. Kiripatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**(13), 3521–3526 (2017). <https://doi.org/10.1073/pnas.1611835114>
10. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning (2019)
11. Mahmud, M., Kaiser, M.S., Hussain, A., Vassanelli, S.: Applications of deep learning and reinforcement learning to biological data. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(6), 2063–2079 (2018)
12. Mermillod, M., Bugaiska, A., Bonin, P.: The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front. Psychol.* **4** (2013)
13. Mnih, V., et al.: Playing atari with deep reinforcement learning. arXiv preprint [arXiv:1312.5602](https://arxiv.org/abs/1312.5602) (2013)
14. Moody, J.E., Saffell, M.: Reinforcement learning for trading systems and portfolios. In: *Knowledge Discovery and Data Mining* (1998)
15. Nguyen, T.H., Shirai, K., Velcin, J.: Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* **42**(24), 9603–9611 (Nov2015). <https://doi.org/10.1016/j.eswa.2015.07.052>, <https://hal.science/hal-01203094>
16. Oliveira, N., Cortez, P., Areal, N.: The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Syst. Appl.* **73** (12 2016). <https://doi.org/10.1016/j.eswa.2016.12.036>
17. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284 (2018)
18. Passalis, N., Tzelepi, M., Tefas, A.: Probabilistic knowledge transfer for lightweight deep representation learning. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(5), 2030–2039 (2020)

19. Pénasse, J.: Understanding alpha decay. *Manage. Sci.* **68**(5), 3966–3973 (2022)
20. Rusu, A.A., et al.: Policy distillation. arXiv preprint [arXiv:1511.06295](https://arxiv.org/abs/1511.06295) (2015)
21. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015)
22. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint [arXiv:1506.02438](https://arxiv.org/abs/1506.02438) (2015)
23. Tran, D.T., Iosifidis, A., Kannianen, J., Gabbouj, M.: Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(5), 1407–1418 (2018)
24. Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from limit order book using convolutional neural networks. In: Proceedings of the IEEE International Conference on Business Informatics (2017). <https://doi.org/10.1109/CBI.2017.23>, iNT=sgn,”Tsantekidis, Avraam”; IEEE International Conference on Business Informatics ; Conference date: 01-01-1900
25. Tsantekidis, A., Passalis, N., Toufa, A.S., Saitas Zarkias, K., Chairistanidis, S., Tefas, A.: Price trailing for financial trading using deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **PP**, 1–10 (06 2020). <https://doi.org/10.1109/TNNLS.2020.2997523>
26. Xu, J., Zhu, Z.: Reinforced continual learning. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
27. Zhang, Z., Zohren, S., Roberts, S.: DeepLOB: deep convolutional neural networks for limit order books. *IEEE Trans. Signal Process.* **67**(11), 3001–3012 (2019). <https://doi.org/10.1109/tsp.2019.2907260>





# Orthogonal Latent Compression for Streaming Anomaly Detection in Industrial Vision

Han Gao<sup>1,2,4</sup> , Huiyuan Luo<sup>2,4</sup> , Fei Shen<sup>1,2,3,4</sup> ,  
and Zhengtao Zhang<sup>1,2,3,4</sup>  

<sup>1</sup> The School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>2</sup> The Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China  
zhengtao.zhang@ia.ac.cn

<sup>3</sup> The Binzhou Institute of Technology, Binzhou 256606, Shandong, China

<sup>4</sup> The CAS Engineering Laboratory for Intelligent Equipment and Technology of Industrial Vision, Beijing 100190, China

**Abstract.** Although existing industrial anomaly detection methods perform well, they are trained on offline datasets collected in advance and remain unchanged once the training is complete. Simultaneously, they assume that the data is static without any drift. However, data in industrial scenarios, especially in sequential assembly lines, usually arrives dynamically in streams and suffers from data drift over time, such as lighting variations and digital noise. The offline training paradigm and inability to dynamically update of existing methods are inconsistent with the data characteristics of streaming dynamics, and it is also difficult to quickly adapt to streaming data drift. To this end, we propose a streaming anomaly detection method that can not only learn dynamically based on the production line data stream, but also adapt to data drift as quickly as possible by effectively utilizing a small amount of drifted training data. The core idea of the proposed method is to compress the features into an orthogonal latent space and constrain the features with nearest reconstruction and maximum separability to maximally capture the normal patterns of the data. Extensive experiments on three real industrial datasets demonstrate our method's excellent performance in stream anomaly detection tasks and rapid adaptability to data drifts. Additionally, our method has lower modeling complexity and higher computational efficiency. It also achieves state-of-the-art performance in offline industrial image anomaly detection and localization tasks. Source code will be released upon paper acceptance.

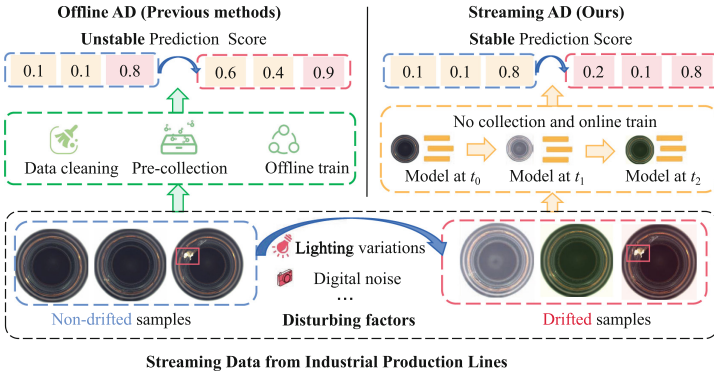
**Keywords:** Anomaly detection · Streaming data · Industrial vision · Orthogonal compression · Unsupervised learning

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_7](https://doi.org/10.1007/978-3-031-78189-6_7).

# 1 Introduction

In industrial vision defect detection tasks, acquiring labeled defect samples is challenging because defects can be extremely small and difficult to collect. Anomaly detection (AD) [16] has gained attention as an effective method to detect defects without the effort of labeling defect samples.

Existing AD methods typically involve two phases: data pre-collection and offline training. In the data pre-collection phase, they often require collecting data from production lines over a period of time and then constructing a training set containing only the normal samples (i.e., defect-free samples). Subsequently, in the offline training phase, AD methods minimize the reconstruction error between the modeled and true normal patterns through techniques such as memory banks [2, 3, 7, 9, 11, 13], normalization flows [6, 21], reconstruction [15, 17–19, 23, 24, 26], and knowledge distillation [4, 25]. For the testing samples, if their patterns deviate from the normal patterns, they are predicted as anomalies (i.e., defect samples). Therefore, existing AD methods generally assume the normal patterns to be statistically static, independent, identically distributed (i.i.d.), and have consistent application scenarios. They do not continue to update the models once training is complete.



**Fig. 1.** In real industrial scenarios, data drift commonly occurs within data streams. Previous anomaly detection methods usually require data cleaning to obtain an offline and drift-free training set. Conversely, our method can rapidly adapt to the drift with a limited amount of data by maximizing the capture of normal data patterns.

However, the assumption that the normal patterns are static and offline can easily be violated in real industrial scenarios. As depicted in Fig. 1, data from production lines typically arrive in a streaming manner. Due to environmental noise and device interference, such as lighting variations and digital noise, data streams often suffer from data drift over time. Samples within a batch of data may exhibit highly varied patterns due to data drift, yet they are all normal. For existing methods, since their normal patterns are modeled based on past data

and the models lack the ability to update, they can easily misclassify drifted samples, for example, considering drifted normal samples as anomalies.

When data drift occurs in the data streams of production lines, it is crucial for AD methods to adjust the model as early as possible based on a small number of newly drifted samples, as this can minimize disruption to the running of the production line. However, most of the existing methods pursue excellent detection performance through high complexity models and the offline learning paradigm. As a result, when data drift occurs, they require a large amount of normal data to sufficiently train the model. For example, they may simulate various potential drift with data augmentation tools to generate vast drifted normal samples or collect training data over long periods on production lines where data drift may occur. Such non-real-time learning paradigms are inefficient. Faced with rapidly dynamic data streams on the production line, AD methods need to adapt to data drift quickly and in real-time. In other words, AD methods need to effectively utilize a small amount of drifted training data to maximize the capture of normal data patterns, thus adapting to data drift as quickly as possible.

For this purpose, this paper proposes a streaming anomaly detection method that maximizes the capture of normal patterns. Compared to existing methods, our method can adapt to the drift in the data stream faster. As illustrated in Fig. 2, the method consists of a pre-trained convolutional neural network (CNN) backbone as an encoder, a learnable  $1 \times 1$  convolutional layer as a projector, and an orthogonal latent space. The projector transfers pre-trained features from the encoder into the orthogonal latent space. Inspired by principal component analysis (PCA) [12], the transferred features need to satisfy two principles: nearest reconstruction and maximum separability. In other words, the distance error between the features and the coordinate basis vectors (which would be introduced in Sec 3.) of the orthogonal latent space should be minimized, and the features of different attributes should be kept as far apart as possible. The model is then optimized based on these two principles to maximize the capture of the normal patterns of data.

Comprehensive experiments are conducted on three real industrial datasets: MVTec AD [1], MPDD [8], and VisA [27]. The proposed method achieves state-of-the-art (SOTA) anomaly detection performance in the streaming scenarios. Furthermore, this paper simulates two types of data drift which are common in industrial environments, including digital noise and lighting variations. The proposed method demonstrates the fastest adaptation capability. Additionally, our method also exhibits lower modeling and spatial complexity, as well as higher computational efficiency. Finally, the proposed method achieves SOTA performance in the offline industrial anomaly detection and localization tasks, which is the setting where previous methods are commonly evaluated.

## 2 Related Work

From the perspective of constructing normal patterns, existing anomaly detection methods can be divided into two categories: normal pattern memory banks

[2, 3, 7, 9, 11, 13] and trainable normal pattern estimators [4, 6, 15, 17–19, 21, 23–26].

The operational process of memory bank methods [2, 3, 7, 9, 11, 13] is as follows: during the training phase, the model extracts features from the training set to construct normal patterns memory bank. The features of the test samples serve as query vectors to match key normal features in the memory bank. In order to reduce the size of the memory bank, PatchCore [13] applies a core-set sampling strategy. CFA [11] designs a k-means based compression scheme. However, when considering the data drift in data streams, these methods face two challenges: (1) the detection performance depends on the integrity of the memory and thus requires extensive normal samples to comprehensively represent normal patterns. The process of re-collecting training data is inefficient and may result in biased memory banks; (2) if the memory banks are dynamically updated based on the drifted data, the memory update mechanisms must be incorporated, yet existing methods such as PatchCore [13] and PaDiM [3] have not yet considered this aspect.

The learnable anomaly detectors [4, 6, 15, 17–19, 21, 23–26] aims to model the normal patterns and assign higher abnormal scores to samples that cannot be correctly modeled. To achieve this, end-to-end architectures such as generative methods are widely adopted. Generative methods use autoencoders [20, 23, 24] or generators [15] to implicitly learning normal patterns through reconstruction tasks. Although these methods [14, 23] provide intuitive and explainable results, they often encounter difficulties in handling objects with complex structures and may also lead to shortcut learning issue, where abnormal regions are correctly reconstructed.

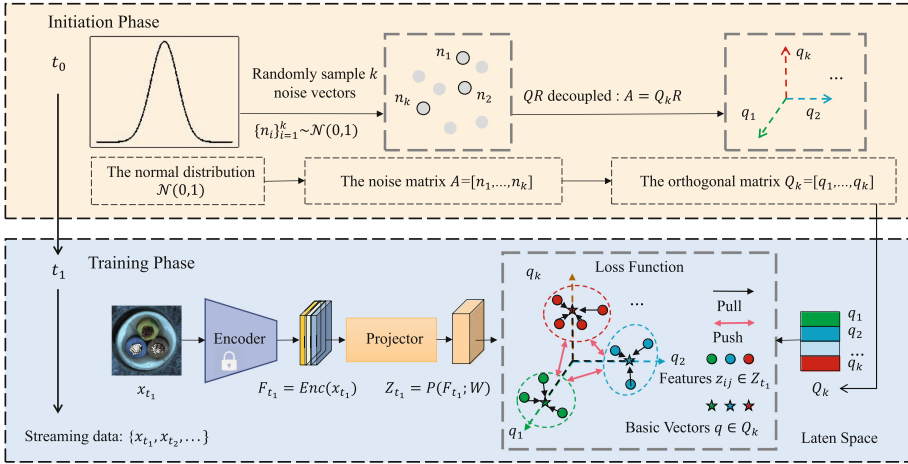
## 3 Method

### 3.1 Overview

As shown in Fig. 2, the architecture of our method consists of three components: a pre-trained and frozen encoder  $Enc$ , a projector  $P$  with learnable parameters  $W$  and an orthogonalized matrix  $Q_k$ .

*Initialization Phase:* Firstly,  $k$  noise vectors with dimension  $d$  are randomly sampled from the normal distribution  $\mathcal{N}(0, 1)$  to form the noise matrix  $\mathbf{A} = [\mathbf{n}_1, \dots, \mathbf{n}_k] \in \mathbb{R}^{k \times d}$ . Then the orthogonal matrix  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k] \in \mathbb{R}^{k \times d}$  is obtained by QR decomposing, i.e.,  $\mathbf{A} = \mathbf{Q}_k \mathbf{R}$ . The  $\mathbf{Q}_k$  are subsequently set as the basis vectors in the latent space.

*Training Phase:* For the streaming data  $\mathbf{x}_t \in \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \dots\}$ , the pre-trained features is extracted by the  $Enc$ :  $\mathbf{F}_t = Enc(\mathbf{x}_t) \in \mathbb{R}^{s \times d}$ , where  $s = w \times h$  represents the spatial resolution and  $d$  denotes the feature dimension. Subsequently, since the ImageNet-trained [5] backbone introduces domain bias when applied to anomaly detection tasks. We use the projector  $P$  to transfer the biased feature  $\mathbf{F}_t$  into  $\mathbf{Z}_t = P(\mathbf{F}_t; W) \in \mathbb{R}^{s \times d}$  to mitigate this bias.



**Fig. 2.** Overview of the proposed method.  $k$  vectors are firstly randomly sampled from the normal distribution  $\mathcal{N}(0,1)$  to form the matrix  $A \in \mathbb{R}^{k \times d}$ . Subsequently, the orthogonal matrix  $Q_k \in \mathbb{R}^{k \times d}$  is obtained by QR decomposing:  $A = Q_k R$ . During the training phase, the pre-trained features  $F_t$  is transferred to  $Z_t$ . The proposed loss function would constrain  $Z_t$  with nearest reconstruction and maximum separability.

Based on the principles of nearest reconstruction and maximum separability, the proposed method designs a loss function to optimize the features  $Z_t$ , aiming to maximize the capture of normal patterns.

*Testing Phase:* During the testing phase, abnormal samples would fail to be projected by the trained  $P$  into the orthogonal latent space. Thus, the distance deviation after projection can be regarded as the abnormal score. Our method’s pseudo-code is shown in Appendix.

### 3.2 Orthogonal Latent Compression

Existing anomaly detection methods use high-complexity models [15, 23] to accurately model normal patterns. These methods lack strict constraints on normal patterns, thus limiting their ability to maximize the capture of normal patterns and forcing them to collect a large number of normal samples to fully cover normal patterns. This can lead to data redundancy and an inability to quickly update the model to adapt to data drift based on a small amount of emerging drifted training data. To this end, inspired by principal component analysis (PCA), we believe that an efficient way to acquire normal patterns is to obtain their decoupled, orthogonal latent patterns. This implies that the captured normal patterns  $Z_t$  should satisfy the principle of nearest reconstruction and maximum separability.

PCA [12] requires the error between the reconstructed samples and the original samples to be minimized. For a sample  $x_t$ , the pre-trained features  $F_t = Enc(x_t) \in \mathbb{R}^{s \times d}$ . The mean  $\mu \in \mathbb{R}^{1 \times d}$  and the covariance matrix

$\mathbf{C} = (\mathbf{F}_t - \boldsymbol{\mu})^T (\mathbf{F}_t - \boldsymbol{\mu}) \in \mathbb{R}^{d \times d}$  are computed. The symmetric matrix  $\mathbf{C}$  can be decomposed as  $\mathbf{C} = \mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T$ , where  $\mathbf{Q} \in \mathbb{R}^{d \times d}$  is an orthogonal matrix whose columns contain principal components, each of which corresponds to an eigenvector of  $\mathbf{C}$ . The first  $k$  columns of  $\mathbf{Q}$ , denoted as  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k] \in \mathbb{R}^{k \times d}$ , represent the directions with the maximum variance (i.e., the Max-Var directions). The PCA transformation  $h$  can be defined as follows:

$$\mathbf{Z}_{pca} = h(\mathbf{F}_t; \boldsymbol{\mu}, \mathbf{Q}_k) = (\mathbf{F}_t - \boldsymbol{\mu})\mathbf{Q}_k \quad (1)$$

$$\mathbf{F}_{recon} = h^{-1}(\mathbf{Z}_{pca}; \boldsymbol{\mu}, \mathbf{Q}_k) = \mathbf{Z}_{pca}\mathbf{Q}_k^T + \boldsymbol{\mu} \quad (2)$$

PCA compresses normal patterns by minimizing the reconstruction error  $\mathcal{L} = \|\mathbf{F}_t - \mathbf{F}_{recon}\|$ . However, PCA is performed on the batches of data, which means that all data must be used to compute  $\mathbf{Q}_k$ . Batch PCA limits the efficiency in streaming scenarios because it must be recalculated when new data arrives. Therefore, we propose a deep method to simulate the data processing process of PCA while overcoming the drawbacks of batch computation and adapting the streaming anomaly detection tasks.

According to Eq. 1, PCA aims to maximize the separability of normal patterns by projecting  $\mathbf{F}_t$  onto the Max-Var directions of the data, i.e., the principal component  $\mathbf{Q}_k$ . These Max-Var directions are decoupled because they correspond to the eigenvectors of different eigenvalues of the covariance matrix  $\mathbf{C}$ . Similarly, our method aims to achieve maximal separability by projecting  $\mathbf{Z}_t$  onto the basis vectors in the matrix  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k]$ . Therefore,  $\mathbf{Q}_k$  should be sufficient decoupled. Furthermore, when considering fine-grained anomaly localization tasks, it is necessary to detect whether each patch-level feature  $\forall \mathbf{z}_t \in \mathbf{Z}_t$  is abnormal. Hence, we do not specify  $\mathbf{Q}_k$  as the eigenvectors of the covariance matrix  $\mathbf{C}$  because this still considers the global distribution of the entire image. Instead,  $\mathbf{Q}_k$  is defined as an orthogonal matrix obtained by QR decomposing a noise matrix  $\mathbf{A} = [\mathbf{n}_1, \dots, \mathbf{n}_k]$  as follows:

$$\mathbf{A} = \mathbf{Q}_k \mathbf{R} \quad (3)$$

For  $\forall \mathbf{z}_t \in \mathbf{Z}_t$ , the projector  $P$  would align them with the nearest basis vector in  $\mathbf{Q}_k$ . Since  $\mathbf{Q}_k$  is orthogonal,  $\mathbf{Z}_t$  will be decoupled. Meanwhile,  $\mathbf{z}_t$  should be far away from the remaining basis vectors in  $\mathbf{Q}_k$ . Upon the completion of training,  $\mathbf{Z}_t$  would become maximally separable since they are projected onto different and orthogonal basis vectors in  $\mathbf{Q}_k$ . Additionally, this process can also be explained by the nearest reconstruction theory of PCA. This is because, the features in  $\mathbf{Z}_t$  would align with the principal components  $\mathbf{Q}_k$  upon training completion, which would minimize the reconstruction error between  $\mathbf{Z}_t$  and  $\mathbf{Q}_k$ .

### 3.3 Loss Function

Based on the above analysis, the projector  $P$  would align  $\mathbf{Z}_t$  with the basis vectors in  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k]$ . The optimization objective is defined as follows:

$$\arg \max sim(\mathbf{Z}_t, \mathbf{Q}_k) \quad (4)$$

**Table 1.** Analysis of the model complexity and the spatial complexity.

Methods	SPADE [2]	PaDiM [3]	PatchCore [13]	CFA [11]	Ours
Modeling Complexity	$\mathcal{O}( \mathcal{X} HWD)$	$\mathcal{O}( \mathcal{X} HWD^2)$	$\mathcal{O}( \mathcal{X} HWD)$	$\mathcal{O}(HWD)$	$\mathcal{O}(KD)$
Spatial Complexity	$\mathbb{R}^{ \mathcal{X}  \times H \times W \times D}$	$\mathbb{R}^{H \times W \times D^2}$	$\mathbb{R}^{ \mathcal{X}  \times \gamma(H \times W) \times D}$	$\mathbb{R}^{\gamma(H \times W \times D)}$	$\mathbb{R}^{K \times D}$

\*  $\mathcal{X}$  is the dataset scale and  $\gamma$  is the compression ratio.  $H, W, D$  are the feature dimensions.

where the  $sim(\cdot)$  is set to the dot product. Since the basis vectors  $\forall \mathbf{q} \in \mathbf{Q}_k$  are decoupled from each other, it is difficult to maximize the similarity between  $\mathbf{Z}_t$  and each basis vector in  $\mathbf{Q}_k$ . Therefore, we calculate the distance between  $\forall \mathbf{q} \in \mathbf{Q}_k$  and  $\forall \mathbf{z}_t(i, j) \in \mathbf{Z}_t$  and sort them. The basis vector  $\mathbf{q}^+$  with the shortest distance is the vector need to be aligned, and the remaining  $k - 1$  basis vectors are the vectors  $\mathbf{q}^-$  need to be moved away. For  $\forall \mathbf{z}_t(i, j) \in \mathbf{Z}_t$ , the process of determining  $\mathbf{q}^+$  and  $\mathbf{q}^-$  is as follows:

$$\mathbf{q}^+ = \arg \min_q \|\mathbf{z}_t(i, j) - \mathbf{q}\|, \mathbf{q}^- = \{\mathbf{q}_i \in \mathbf{Q}_k : \mathbf{q}_i \notin \mathbf{q}^+\}, \forall \mathbf{q} \in \mathbf{Q}_k \quad (5)$$

The loss function  $\mathcal{L}$  could be defined in Eq. 6.

$$\mathcal{L} = \sum_{\mathbf{z}_t(i, j) \in \mathbf{Z}_t} -\log \frac{\sum_{\mathbf{q}^+} \exp(sim(\mathbf{z}_t(i, j), \mathbf{q}^+))}{\sum_{\mathbf{q}^+} \exp(sim(\mathbf{z}_t(i, j), \mathbf{q}^+)) + \sum_{\mathbf{q}^-} \exp(sim(\mathbf{z}_t(i, j), \mathbf{q}^-))} \quad (6)$$

As shown in Table 1, our method has the lowest modeling and spatial complexity, only related to the number of basis vectors in  $\mathbf{Q}_k$  and the feature dimension  $d$ .

### 3.4 Abnormal Scores

After the training is completed, the projector  $P$  could map  $\mathbf{Z}_t$  to the orthogonalized basis vectors in  $\mathbf{Q}_k$ . However, for abnormal samples, the model is unable to accomplish this mapping. Therefore, the anomaly score can be calculated based on the distance between the testing features and  $\mathbf{Q}_k$ . Suppose the testing sample is  $\mathbf{x}_{test}$ , its transferred features are  $\mathbf{Z}_{test}$ . For  $\forall \mathbf{z}_{test}(i, j) \in \mathbf{Z}_{test}$ , we take its shortest distance from  $\mathbf{Q}_k$  as the anomaly degree of the testing sample deviation from normal patterns, denoted as  $d_{test}(i, j)$ :

$$d_{test}(i, j) = \min \|\mathbf{z}_{test}(i, j) - \mathbf{q}\|, \forall \mathbf{q} \in \mathbf{Q}_k \quad (7)$$

The anomaly score map  $\mathcal{A} \in \mathbb{R}^{w \times h}$  is defined in Eq. 8.

$$\mathcal{A}(i, j) = \frac{d_{test}(i, j)}{\sum_{i, j} d_{test}(i, j)} \quad (8)$$

Subsequently,  $\mathcal{A}$  is upsampled until it matches the size of  $\mathbf{x}_{test}$ . The detection accuracy of the model is computed based on the ground truth mask of  $\mathbf{x}_{test}$  and the predicted anomaly score map  $\mathcal{A}$ .

## 4 Experiments

### 4.1 Experimental Setup

**Implementation Details.** Our method is evaluated on three real industrial datasets, include MVTec AD [1], MPDD [8] and VisA [27]. The compared methods include four SOTA anomaly detection methods: PatchCore [13], CFA [11], FastFlow [21], and DRAEM [23]. The optimizer uses Adam [10] with a base learning rate of 1e-3 and weight decay of 5e-4. The input images are resized to  $256 \times 256$  and then centrally cropped to  $224 \times 224$  and normalized. The backbone selects the Wide-ResNet50 [22], which is trained on the ImageNet dataset [5]. The combination of layers 2, 3 and 4 of the backbone is utilized as the pre-trained features  $\mathbf{F}_t$ . With the batch size  $b$ , layers 2, 3 and 4 are  $b \times 256 \times 56 \times 56$ ,  $b \times 512 \times 28 \times 28$  and  $b \times 1024 \times 14 \times 14$ , respectively. *Layer 3* and *Layer 4* are upsampled to  $56 \times 56$  size. These features are combined as  $F_t = [\textit{Layer 2, 3, 4}] \in \mathbb{R}^{b \times 1792 \times 56 \times 56}$ . The projector  $P$  is a  $1 \times 1$  convolutional layer. The noise matrix  $\mathbf{A}$  is randomly sampled from the normal distribution  $\mathcal{N}(0, 1)$ . The  $\mathbf{Q}_k$  is obtained by  $\mathbf{A} = \mathbf{Q}_k \mathbf{R}$  and contains  $k = 10$  orthogonalized basis vectors.

**Protocols.** Denote the training set and the testing set as  $\mathcal{X}_{train}$  (contains  $t$  normal samples) and  $\mathcal{X}_{test}$  (contains both normal and anomalous samples), respectively. We simulate the streaming scenario according to the following setup. During the training phase, we specify that only one training sample  $\mathbf{x}_t \in \mathcal{X}_{train}$  arrives at each time step. The anomaly detection method can only see  $\mathbf{x}_t$  at a time and uses it to update the model. To evaluate the ability of the anomaly detection method to capture normal patterns, we evaluate the detection performance on the testing set  $\mathcal{X}_{test}$  after learning each training sample  $\mathbf{x}_t$ . The evaluation metrics are the image-level AUROC (I-AUROC) and pixel-level AUROC (P-AUROC), respectively. By tracking the real-time changes of the I-AUROC and P-AUROC metrics after learning each training sample (total  $t$  normal samples which arrives as streams), we can observe the changing relationship between the detection performance and the number of input training samples (from 0 to  $t$ ). This relationship reflects the ability of the model to learn quickly on the training samples that arrive in streams—if the models are able to detect well after receiving fewer streaming samples, then they have the ability to learn and adapt quickly, i.e., they are able to maximize the ability to capture the normal patterns of the data (As the result shown in Fig. 3). In addition, since most existing methods are designed for offline scenarios, we modify them to suit streaming scenarios, thus ensuring a comprehensive and fair comparison with our method. Specifically, exponential moving average (EMA) is incorporated into the memory banks construction of PatchCore and CFA, enabling these methods to dynamically update the memory banks with stream data input. The pseudo codes are shown in Appendix.



## 4.2 Evaluation on Streaming AD

**Accuracy Report.** Table 2 presents the average I-AUROC and P-AUROC metrics on the MVTec AD, MPDD and VisA datasets. Table 3 presents the results on each category of MVTec AD. The detailed results on MPDD and VisA can be found in Appendix. For the I-AUROC metric, our method outperforms others by 0.6% and 1.9% on the MVTec AD and VisA datasets, respectively. For the P-AUROC metric, our method outperforms on all three datasets.

**Table 2.** Streaming performance.  $\cdot/\cdot$  denotes I-AUROC  $\uparrow$  and P-AUROC  $\uparrow$ .

Method	PatchCore [13]	CFA [11]	FastFlow [21]	DRAEM [23]	Ours
MVTec AD	0.872/0.934	0.966/0.975	0.927/0.967	0.791/0.730	<b>0.972/0.976</b>
MPDD	0.721/0.954	0.865/0.977	<b>0.907</b> /0.885	0.719/0.727	0.874/ <b>0.978</b>
VisA	0.801/0.946	0.924/0.981	0.883/0.965	0.712/0.561	<b>0.943/0.987</b>

**Table 3.** Streaming performance on MVTec AD dataset.

Metric	I-AUROC $\uparrow$					P-AUROC $\uparrow$				
	Method	PatchCore	CFA	FastFlow	DRAEM	Ours	PatchCore	CFA	FastFlow	DRAEM
Bottle	0.949	<b>1.000</b>	<b>1.000</b>	0.857	<b>1.000</b>	0.953	0.981	0.978	0.610	<b>0.982</b>
Cable	0.897	0.914	<b>0.940</b>	0.557	0.926	0.937	0.965	0.945	0.541	<b>0.966</b>
Capsule	0.753	0.937	0.915	0.552	<b>0.950</b>	0.967	0.987	0.982	0.657	<b>0.988</b>
Carpet	0.990	<b>1.000</b>	0.987	0.907	<b>1.000</b>	0.991	0.991	0.982	0.882	<b>0.992</b>
Grid	0.805	0.912	<b>0.967</b>	0.874	0.943	0.738	0.922	<b>0.962</b>	0.805	0.938
Hazelnut	0.949	<b>1.000</b>	0.771	0.833	<b>1.000</b>	0.953	0.984	0.959	0.814	<b>0.986</b>
Leather	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.960	<b>1.000</b>	0.993	0.992	<b>0.996</b>	0.723	0.992
Metalnut	0.681	0.996	0.970	0.746	<b>1.000</b>	0.904	<b>0.990</b>	0.966	0.687	<b>0.990</b>
Pill	0.842	<b>0.976</b>	0.919	0.831	0.964	0.959	<b>0.991</b>	0.974	0.598	0.990
Screw	0.570	<b>0.850</b>	0.739	0.515	0.849	0.891	0.978	0.927	0.933	<b>0.979</b>
Tile	<b>0.997</b>	0.996	0.953	0.965	<b>0.997</b>	0.957	<b>0.964</b>	0.943	0.766	0.963
Toothbrush	0.936	0.975	0.828	0.689	<b>1.000</b>	0.973	0.986	0.978	0.829	<b>0.988</b>
Transistor	0.774	0.953	<b>0.990</b>	0.746	0.955	0.869	<b>0.951</b>	0.950	0.582	<b>0.951</b>
Wood	0.996	<b>1.000</b>	0.965	0.955	0.999	0.950	0.952	<b>0.981</b>	0.792	0.951
Zipper	0.945	0.988	0.960	0.871	<b>0.995</b>	0.975	<b>0.986</b>	0.983	0.731	<b>0.986</b>
Average	0.872	0.966	0.927	0.791	<b>0.972</b>	0.934	0.975	0.967	0.730	<b>0.976</b>

**Convergence Speed.** All methods are evaluated on the ability to capture normal patterns of data. To this end, we observe which method could achieve higher detection accuracy faster when receiving an equal amount of streaming data. As shown in Fig. 3, our method quickly achieves 90% I-AUROC using 20% data, and ultimately maintains a high accuracy. In contrast, PatchCore stops at a

lower accuracy after a slow rise. This is due to the fact that coreset of PatchCore is not representative of the true normal patterns which is dynamically changing, as it is not feasible to pre-collect the dataset at the initial phase. In addition, CFA also shows good performance, mainly due to its ability to adapt image features to new tasks. However, there is still room for improving its detection accuracy since it cannot obtain large-scale datasets in advance to build a high-quality memory bank.

**The Adaptability to Data Drift.** As shown in Fig. 4, we assume that there is no drift in the input training samples from time 0 to  $t$ . The  $t$  is the number of training samples for each category. For example, the  $t$  for the bottle category of the MVTEC AD dataset is 209 because it has 209 training samples. From time  $t + 1$  onwards, the training samples experience drifts  $\mathcal{H}(\cdot)$ , forming  $\hat{\mathbf{x}}_{train}(t + 1) = \mathcal{H}(\mathbf{x}_{train}(t + 1))$ ,  $\hat{\mathbf{x}}_{train}(t + 2) = \mathcal{H}(\mathbf{x}_{train}(t + 2))$ , .... The same drift is applied to  $\mathcal{X}_{test}$ . For example, at time  $t + 1$ , for  $\forall \mathbf{x}_{test} \in \mathcal{X}_{test}$ , it is altered to  $\hat{\mathbf{x}}_{test} = \mathcal{H}(\mathbf{x}_{test})$ . Subsequently, at each time step  $t + 1, t + 2, \dots$ , we evaluate the model on the drifted testing set.

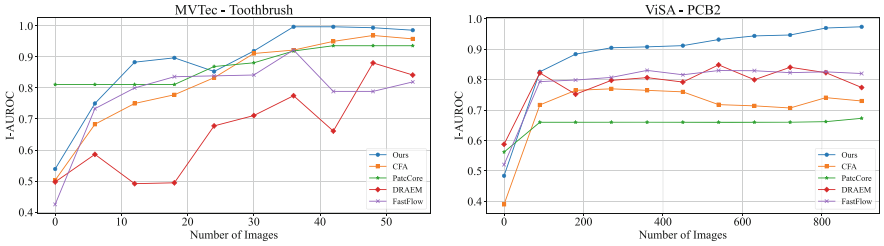
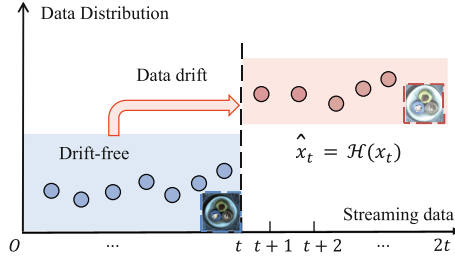


Fig. 3. I-AUROC curve for different methods on two datasets.

Specifically, two types of data drift are introduced, including lighting variations and digital noise. We select one type of drift each time and apply it to the input data  $\mathbf{x}_t$ . The scenario without data drift is set as the baseline, i.e., all methods would firstly train based on  $t$  normal samples  $\mathbf{x}_{train}(0), \dots, \mathbf{x}_{train}(t) \in \mathcal{X}_{train}$ . When data drift occurs, all methods continue learning on the drifted training samples  $\hat{\mathbf{x}}_{train}(t + 1), \dots, \hat{\mathbf{x}}_{train}(2t)$ . This means that all methods learn the number  $t$  of drifted training samples again in a streaming manner. Subsequently, the testing set is subjected to the same drift and used for evaluating.

Table 4 presents the reports on the MVTEC AD dataset. When data drift occurs, all methods experience varying degrees of accuracy degradation. However, our method exhibits better robustness and rapid adaptation to data drift. For lighting variations drift, our method’s I-AUROC metric decreased by -1.1%, while other methods decreased by -10.5%, -7.1%, -5.4%, and -9.6%, respectively. A similar trend can be observed for digital noise: ours: -2.0% vs other methods: -14.8%, -8.2%, -2.1% and -6.1%.



**Fig. 4.** The data stream is assumed not to have any drift during the 0- $t$  period. In time after  $t + 1$ , drift occurs.

**Computational Efficiency.** The computational efficiency of all methods are as shown in Table 5. The evaluation includes throughput per second (TPS), the time for feature extraction and anomaly detection. We conduct five runs for each method, each processing 2000 images from the data stream, and then calculate the average metrics. Our method demonstrates the highest throughput with 187.1 img/s, and the average anomaly detection time is the lowest at 0.7 ms. This excellent efficiency can be attributed to the lower modeling and spatial complexity of our method.

**Visualization of Defect Detection Results.** Defect detection results in MVTec AD dataset are visualized. Figure 5 shows the samples, ground truths and anomaly maps for each category. Despite data drift including lighting variations and digital noise occurs, our method is still able to accurately locate defects.

**Table 4.** Performance comparison with drifted data.

Data drift	Metric	Patchcore[13]	CFA[11]	FastFlow[21]	DRAEM[23]	Ours
None	I-AUROC $\uparrow$	0.872/-0.000	0.966/-0.000	0.927/-0.000	0.791/-0.000	<b>0.972</b> /-0.000
	P-AUROC $\uparrow$	0.934/-0.000	0.975/-0.000	0.967/-0.000	0.730/-0.000	<b>0.976</b> /-0.000
Brightness	I-AUROC $\uparrow$	0.767/-0.105	0.895/-0.071	0.873/-0.054	0.695/-0.096	<b>0.961</b> /-0.011
	P-AUROC $\uparrow$	0.865/-0.069	<b>0.965</b> /-0.010	0.930/-0.037	0.606/-0.124	0.963/-0.013
Gaussian	I-AUROC $\uparrow$	0.724/-0.148	0.884/-0.082	0.906/-0.021	0.730/-0.061	<b>0.952</b> /-0.020
	P-AUROC $\uparrow$	0.855/-0.079	0.961/-0.014	0.934/-0.033	0.601/-0.129	<b>0.966</b> /-0.010

**Table 5.** Computation efficiency.

Methods	PatchCore [13]	CFA [11]	FastFlow [21]	DRAEM [23]	Ours
TPS $\uparrow$	98.1	104.7	60.0	39.6	<b>187.1</b>
Encoder $\downarrow$	6.1	5.2	<b>2.3</b>	3.6	4.6
Detection $\downarrow$	4.1	4.3	16.4	21.6	<b>0.7</b>

\* TPS: Throughput per second *img/s*. Encoder and detection in *ms*.

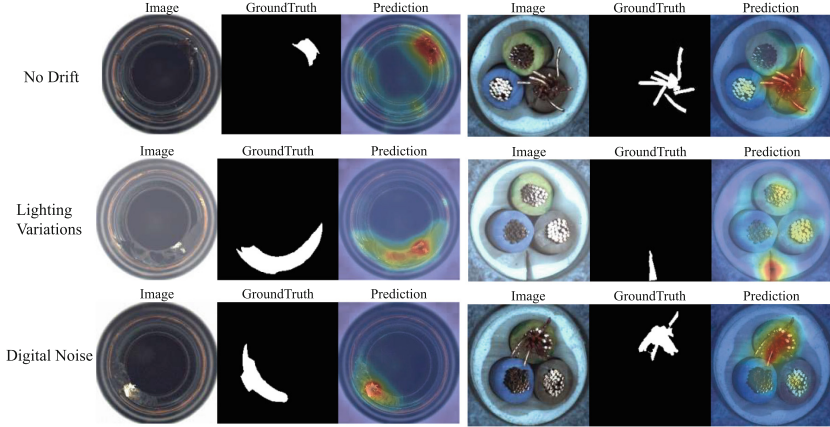


Fig. 5. Visualization of defect detection in MVTec AD dataset.

### 4.3 Evaluation on Offline AD

The performance on offline scenarios is shown in Table 6. Our methods achieve comparable accuracy to the SOTA methods on the full dataset. Moreover, in industrial scenarios, the performance in few-shot scenarios is of great importance. We evaluate the performance of the methods for this issue. Five independent repeated experiments are conducted. When the number of training samples is set as  $N = 1$ ,  $N = 2$ ,  $N = 4$  and  $N = 8$ , our method achieves better performance. Particularly, when  $N = 1$  and  $N = 2$ , our method’s I-AUROC metric exceeded other methods by 4.2% and 4.0%, respectively.

Table 6. Performance comparison under the offline scenario. I-AUROC  $\uparrow$  is reported.

Method	Full Dataset			Few-shot (MVTec AD)			
	VisA	MVTec AD	MPDD	N=1	N=2	N=4	N=8
Patchcore [13]	0.951	<b>0.992</b>	<b>0.948</b>	0.619	0.721	0.817	0.864
CFA [11]	0.920	0.980	0.923	0.813	0.839	0.879	<b>0.923</b>
FastFlow [21]	0.822	0.905	0.887	0.552	0.552	0.729	0.801
DRAEM [23]	0.887	0.981	0.941	0.685	0.777	0.820	0.883
Ours	<b>0.955</b>	<b>0.992</b>	0.922	<b>0.855</b> $\pm$ 0.0111	<b>0.879</b> $\pm$ 0.0130	<b>0.887</b> $\pm$ 0.0007	0.908 $\pm$ 0.0087

### 4.4 Ablation Analysis

The core idea of the proposed method is to transfer the pre-trained features  $F_t$  of normal samples into an orthogonalized latent space  $Q_k$  to obtain  $Z_t = P(F_t; W)$ . In the following, we demonstrate the influence of key factors of the proposed method, including (1) whether it is necessary to constrain  $Q_k$  to be an orthogonalized matrix, and (2) the influence of the number of base vectors  $k$ .

**Orthogonalization Constraint.**  $\mathbf{Q}_k$  is set as different variants. Firstly,  $\mathbf{Q}_k$  is set as the noise matrix  $\mathbf{A}$  which is generated by randomly sampled  $k$  noise vectors from the normal distribution  $\mathcal{N}(0, 1)$ , i.e.,  $\mathbf{Q}_k = \mathbf{A} \sim \mathcal{N}(0, 1)$ . Since the base vectors in  $\mathbf{Q}_k$  are not orthogonalized, we denote this as *None* in Table 7. The second setting of the  $\mathbf{Q}_k$  is applying QR decomposition to the matrix  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{Q}_k \mathbf{R}$ . Thus the  $\mathbf{Q}_k$  becomes an orthogonal matrix. We denote this case with the  $\checkmark$ . The proposed method is evaluated under such two settings. As shown in Table 7, when  $\mathbf{Q}_k$  is not an orthogonal matrix ( $\mathbf{Q}_k = \mathbf{A} \sim \mathcal{N}(0, 1)$ ), there is a noticeable degradation in the detection performance. For the MVTec AD, MPDD, and VisA datasets, the I-AUROC metrics are 77.8%, 62.7%, and 70.6%, respectively. However, when  $\mathbf{Q}_k$  is orthogonalized ( $\mathbf{A} = \mathbf{Q}_k \mathbf{R}$ ), the detection performance improves significantly, corresponding to 97.2%, 87.4%, and 93.4%, respectively. This emphasizes the important impact of the orthogonalization property of  $\mathbf{Q}_k$  on the detection accuracy, i.e., by transferring  $\mathbf{F}_t$  to  $\mathbf{Z}_t$  and constraining  $\mathbf{Q}_k$  decoupled, one can obtain compact normal modes that simultaneously satisfy the nearest reconstruction and maximum separability.

**Table 7.** Orthogonalization constraint of the basis vectors in the matrix  $\mathbf{Q}_k$ .

Orthogonalization	Metric	MVTec AD	MPDD	VisA
None ( $\mathbf{Q}_k \sim \mathcal{N}(0, 1)$ )	I-AUROC $\uparrow$	0.778	0.627	0.706
	P-AUROC $\uparrow$	0.766	0.796	0.812
$\checkmark$ ( $\mathbf{A} = \mathbf{Q}_k \mathbf{R}$ )	I-AUROC $\uparrow$	<b>0.972</b>	<b>0.874</b>	<b>0.934</b>
	P-AUROC $\uparrow$	<b>0.976</b>	<b>0.978</b>	<b>0.987</b>

**Basis Vector Quantity.** The impact of the number  $k$  of base vectors in  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k]$  on the detection performance is evaluated. The detection performance of the proposed method is evaluated under both streaming detection and offline detection scenarios when  $k = 1, 2, 5, 10, 20, 40, 80$ . When  $k = 1$ , the proposed method degrades into a standard one-class detection method, where the pre-trained features  $\mathbf{F}_t$  are aligned to a single base vector after being transferred through the projector. When  $k > 1$ , the proposed method aligns the features  $\mathbf{Z}_t$  to the nearest base vector  $\mathbf{q} \in \mathbf{Q}_k$ . As shown in Table 10, when  $k = 1$ , the I-AUROC metric of the proposed method is 93.4%. As  $k$  increases, the detection performance gradually improves, for example:  $k : 1 \rightarrow 5 \rightarrow 10$ , the corresponding changes in I-AUROC metric are: 93.4%  $\rightarrow$  95.6%  $\rightarrow$  97.2%. However, when  $k$  exceeds 10, performance remains almost unchanged. This is because images typically contain multiple local structures, a powerful prior knowledge. These different local structures encompass varying semantic information and thus require differential treatment. In our method, the number  $k$  can be considered as the process of aligning features with different local structures. Consequently, setting  $k = 1$  forces all features to align with a single basis vector, resulting in

coarse structural descriptions that degrade performance. In contrast, increasing the number  $k$  helps align different features with different basis vectors, conforming to the prior knowledge of local structures. Furthermore, since the model can automatically select the most similar basis vector for each feature, an excessively large  $k$  will not further enhance model performance, as it exceeds the number of different local structures in the image. Experimental results indicate that  $k = 10$  adequately covers the local structure information in the images, and thus is set as the default value. This indicates that when receiving sequentially arriving streaming data in equal amounts, the proposed method indeed helps the anomaly detector to capture the normal data patterns to the maximum extent by aligning the pre-trained features  $\mathbf{F}_t$  to multiple decoupled base vectors  $\mathbf{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k]$ . Additionally, in the offline case, the anomaly detector can converge to a locally optimal normal pattern after multiple epochs of training without considering local structural similarities.

**Backbone.** We conduct ablation experiments about the pre-trained backbone types. As shown in Table 8, our method shows excellent performance across different backbones.

**Table 8.** I-AUROC  $\uparrow$  of our method and FRE.

Backbone	ResNet18	ResNet50	VGG16	EfficientNet-B5	WideResNet50
Ours	0.978	0.988	0.976	0.983	<b>0.992</b>

Additionally, ablation experiments for different layers are shown in Table 9. The proposed method achieves the best accuracy when we combine 2, 3, and 4 layers of features, which means that the middle and high level features are crucial.

**Table 9.** Ablation experiments on different layers.

Layer	(Layer 2)	(Layer 3)	(Layer 4)	(Layer 2, Layer 3)	(Layer 2, 4)	(Layer 3, 4)	(Layer 2, 3, 4)
I-AUROC	0.886	0.968	0.982	0.969	0.987	0.990	<b>0.992</b>

**Table 10.** Impact of memory bank size for performance. Evaluated on MVTec AD.

Scenario	Metric	$k = 1$	$k = 2$	$k = 5$	$k = 10$	$k = 20$	$k = 40$	$k = 80$
Streaming	I-AUROC $\uparrow$	0.934	0.959	0.956	<b>0.972</b>	0.970	0.969	<b>0.972</b>
	P-AUROC $\uparrow$	0.956	0.965	0.961	<b>0.976</b>	0.972	0.973	0.973
Offline	I-AUROC $\uparrow$	0.989	0.990	0.991	<b>0.992</b>	0.991	0.991	0.991
	P-AUROC $\uparrow$	0.980	0.980	<b>0.981</b>	0.980	<b>0.981</b>	0.980	<b>0.981</b>

## 5 Conclusion

In this paper, we propose a streaming anomaly detection method, which carries significant and realistic application value in industrial scenarios, especially in process manufacturing scenarios where data drift is common in data streams. The proposed method can adapt faster than previous methods when data drift occurs in data streams from industrial production lines. Such fast adaptability can be attributed to that the proposed method can capture normal patterns fully with the orthogonal latent compression on a limited number of drifted samples. The proposed method also achieves SOTA performance on offline industrial image anomaly detection and localization tasks. Additionally, the proposed method has lower modeling and spatial complexity, as well as higher computational efficiency.

**Acknowledgements.** This work is supported by, the National Natural Science Foundation of China (U21A20482, 62303461, 62303458), and Beijing Municipal Natural Science Foundation, China, under Grant L243018.

## References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTEC AD—a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
2. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint [arXiv:2005.02357](https://arxiv.org/abs/2005.02357) (2020)
3. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDiM: a patch distribution modeling framework for anomaly detection and localization. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV, pp. 475–489. Springer (2021)
4. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9727–9736 (2022). <https://doi.org/10.1109/CVPR52688.2022.00951>
5. 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)
6. Gudovskiy, D., Ishizaka, S., Kozuka, K.: CFLOW-AD: real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 98–107 (2022)
7. Jang, J., Hwang, E., Park, S.H.: N-Pad: neighboring pixel-based industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4364–4373 (2023)
8. Jezek, S., Jonak, M., Burget, R., Dvorak, P., Skotak, M.: Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In: 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 66–71. IEEE (2021)

9. Jiang, X., et al.: SoftPatch: unsupervised anomaly detection with noisy data. *Adv. Neural. Inf. Process. Syst.* **35**, 15433–15445 (2022)
10. 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, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). <http://arxiv.org/abs/1412.6980>
11. Lee, S., Lee, S., Song, B.C.: CFA: coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access* **10**, 78446–78454 (2022)
12. Ndiour, I., Ahuja, N.A., Genc, E.U., Tickoo, O.: FRE: a fast method for anomaly detection and segmentation. In: 34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023. *BMVA* (2023). <https://papers.bmvc2023.org/0614.pdf>
13. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14318–14328 (2022)
14. Ruff, L., et al.: Deep one-class classification. In: International Conference on Machine Learning, pp. 4393–4402. PMLR (2018)
15. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Niethammer, M., et al. (eds.) *Information Processing in Medical Imaging: 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, pp. 146–157. Springer International Publishing, Cham (2017). [https://doi.org/10.1007/978-3-319-59050-9\\_12](https://doi.org/10.1007/978-3-319-59050-9_12)
16. Tao, X., Gong, X., Zhang, X., Yan, S., Adak, C.: Deep learning for unsupervised anomaly localization in industrial images: a survey. *IEEE Trans. Instrum. Meas.* **71**, 1–21 (2022). <https://doi.org/10.1109/TIM.2022.3196436>
17. Zavrtnik, V., Kristan, M., Skočaj, D.: DSR – a dual subspace re-projection network for surface anomaly detection. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pp. 539–554. Springer Nature Switzerland, Cham (2022). [https://doi.org/10.1007/978-3-031-19821-2\\_31](https://doi.org/10.1007/978-3-031-19821-2_31)
18. Yan, Y., Wang, D., Zhou, G., Chen, Q.: Unsupervised anomaly segmentation via multilevel image reconstruction and adaptive attention-level transition. *IEEE Trans. Instrum. Meas.* **70**, 1–12 (2021). <https://doi.org/10.1109/TIM.2021.3107586>
19. Yang, M., Wu, P., Feng, H.: MemSeg: a semi-supervised method for image surface defect detection using differences and commonalities. *Eng. Appl. Artif. Intell.* **119**, 105835 (2023)
20. You, Z., et al.: A unified model for multi-class anomaly detection. *Adv. Neural. Inf. Process. Syst.* **35**, 4571–4584 (2022)
21. Yu, J., et al.: FastFlow: unsupervised anomaly detection and localization via 2D normalizing flows. *arXiv preprint arXiv:2111.07677* (2021)
22. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference 2016. British Machine Vision Association (2016)
23. Zavrtnik, V., Kristan, M., Skočaj, D.: DRAEM—a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8330–8339 (2021)



24. Zhou, K., et al.: Encoding structure-texture relation with P-net for anomaly detection in retinal images. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX, pp. 360–377. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58565-5\\_22](https://doi.org/10.1007/978-3-030-58565-5_22)
25. Zolfaghari, M., Sajedi, H.: Unsupervised anomaly detection with an enhanced teacher for student-teacher feature pyramid matching. In: *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1–4. IEEE (2022)
26. Zong, B., et al.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: *International Conference on Learning Representations* (2018)
27. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: SPot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022: 17th European Conference*, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX, pp. 392–408. Springer Nature Switzerland, Cham (2022). [https://doi.org/10.1007/978-3-031-20056-4\\_23](https://doi.org/10.1007/978-3-031-20056-4_23)



# Out-of-Distribution Forgetting: Vulnerability of Continual Learning to Intra-class Distribution Shift

Liangxuan Guo<sup>1,2</sup> , Yang Chen<sup>1</sup> , and Shan Yu<sup>1,2,3</sup>  

<sup>1</sup> Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China  
shan.yu@nlpr.ia.ac.cn

<sup>2</sup> School of Future Technology, University of Chinese Academy of Sciences (UCAS),  
Beijing, China

<sup>3</sup> School of Artificial Intelligence, University of Chinese Academy of Sciences  
(UCAS), Beijing, China

**Abstract.** Continual learning (CL) is a key technique enabling neural networks to acquire new tasks while retaining efficiency in previous ones. Standard CL tests revisit old tasks after learning, assuming stable data distribution, which is often impractical. Meanwhile, it is well known that the out-of-distribution (OOD) problem will severely impair the ability of networks to generalize. Rare research considered the influence of CL on the generalizing ability of neural networks. Our research highlights a special form of catastrophic forgetting raised by the OOD problem in CL settings. Through continual image classification experiments, we discovered that: introducing a tiny intra-class distribution shift within a specific category significantly impairs the recognition accuracy of many CL methods. We named it out-of-distribution forgetting (OODF). Moreover, the performance degradation caused by OODF is special for CL, as the same level of distribution shift had only negligible effects in the joint learning scenario. We verified that most CL strategies except for parameter isolation ones are vulnerable to OODF. Taken together, our work identified an under-attended risk during CL, highlighting the importance of developing approaches that can overcome OODF. Code available: <https://github.com/Hiroid/OODF>.

**Keywords:** Deep learning · Continual Learning · Out-of-Distribution Forgetting · Catastrophic Forgetting

## 1 Introduction

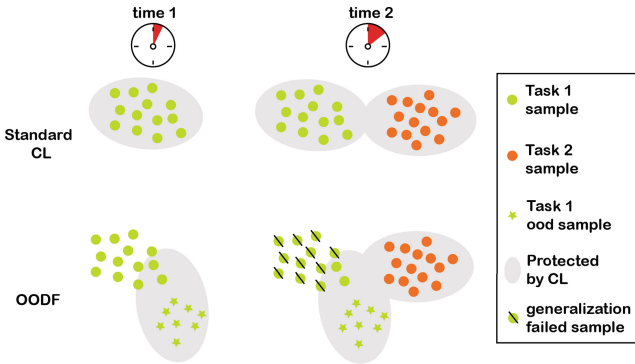
Learning models based on artificial neural networks usually suffer from catastrophic forgetting (CF) [3, 17, 21] in open environments. Researchers have pro-

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_8](https://doi.org/10.1007/978-3-031-78189-6_8).

posed various continual learning (CL) methods for deep neural networks to overcome CF. These include strategies based on parameter regularization, memory replay, and parameter isolation etc. [7, 10, 26]. By enabling a system to learn new tasks and maintain its performance on old tasks, CL has made significant progress in incremental image recognition and other computer vision tasks [34, 35].

Even with great advances, the current CL strategies may still not cope well with the problem of CF in real world. One of the many concerns is the noise tolerance of CL strategies. In the review [11], the authors concerns that continual learning machines may not perform well if there is a large distribution shift between the data encountered in the inference phase and those in the training phase. In the present manuscript, we surprisingly find that the practical situation is much more severe. Our work indicates that even a tiny intra-class distribution shift, negligible to human observers, can introduce severe performance impairments for current CL methods.



**Fig. 1.** Illustration of out-of-distribution forgetting. There are two continual learning scenarios, the top row is a standard continual learning paradigm, while the bottom row is a continual learning paradigm with an intra-class distribution shift on task 1. At time 1 in the OODF paradigm, although the generalization of task 1 was equally good compared to the standard CL setting, the protection provided by CL methods mainly focuses on out-of-distribution samples of task 1, leading to severe deficits in performing task 1 after learning task 2.

Specifically, we named this phenomenon out-of-distribution forgetting (OODF) (Fig. 1). This phenomenon is special for CL as the performance degradation is caused by the subsequent learning of other categories, which is different from the well-studied OOD problem in the setting of joint learning [28]. We believe OODF is an important yet under-attended problem for developing as well as evaluating CL methods in the future because:

- OODF is commonly present in various CL strategies and settings. We verified its existence in both regularization-based and memory-based CL strategies on different tasks with different network structures.

- OODF is elusive and challenging to detect. Its effects don’t manifest immediately post-training on shifted data but rather emerge as the model learns new tasks, reflecting its nature as a unique form of CF. Thus, a continual learning machine affected by OODF can be considered capable of performing certain tasks but it actually will fail. In addition, it only affects the class contaminated with the distribution-shifted data, without influencing other tasks.
- OODF can be triggered by various conditions leading to distribution-shifted data. We find that OODF severely impairs the CL performance, regardless of the approach causing the shift (local or global perturbation), as well as the reason behind it (deliberately designed attack or accident).
- Preliminary findings indicate that introducing a rejection category will help to alleviate OODF.

Our work identified OODF as a specific form of CF barely covered in previous studies, which is an important issue to consider for improving the security and robustness of the CL methods towards their application in practical circumstances.

## 2 Related Works

### 2.1 Continual Learning

In recent years, various algorithms have been proposed to overcome the CF problem in CL tasks. Although new technique and applications such as prompt-based CL [22] and continual pre-training (CPT) [16,37] etc. are getting noticed, the components underlying can still be decomposed as these essential strategies: parameter regularization strategy [5,36], memory replay strategy [1,20,23,25,29], and parameter isolation (also known as architecture-based) strategy [12].

This work focuses on the class incremental scenario [8,33], as it is a real scene where CL models need to identify all classes (i.e. categories) without task IDs. Meanwhile, we allow models to train each task offline (as opposed to online CL [15]), ensuring a better performance on standard CL, as a higher baseline for subsequent OODF experiments.

### 2.2 Security Concerns of Neural Networks

The first concern addresses the **OOD** problem in neural networks. In non-CL paradigms, significant loss of generalization occurs if there’s a shift between training and testing datasets, especially due to corruption or perturbation [27,28]. The second concern centers on the security of well-trained neural networks, particularly against deliberate attacks like **adversarial** [24], **data poisoning** [9] and **backdoor attacks** [14]. These areas mentioned above comprehensively investigate the security problem in different stages, purposes, and means. However, most models in this area are static, employing un-sequential joint training procedures. Notably, few studies have focused on the unique security and robustness risks inherent to CL.

### 2.3 Several Concerns of Continual Learning

The first concern was security in CL. Security of neural networks and CL have been studied largely in parallel until recently. Guo et al. [6] propose the GREV method to attack the A-GEM methods with adversarial samples and disseminate misinformation in the memory buffer. Umer et al. [30–32] show that it is possible to attack CL by modifications on both training samples and labels to give a misleading supervising signal. Li and Ditzler [13] attack several parameter regularization strategies by injecting poisoned adversarial samples into subsequent tasks following the target task, in the task incremental scenario. However, they implement targeted poisoning attacks by injecting poisoned adversarial samples into subsequent non-target tasks.

The second concern was the real-world application of CL. Recent studies highlight that continual agents, when exposed to out-of-distribution samples in open-world settings, may compromise safety and performance. Caccia et al. [2] define learning new tasks as the OOD problem (compared to the learned old tasks), a perspective distinct from OODF, which presents unique challenges and definitions. Mundt et al. [18, 19] conducted experiments on *reverse continual learning*. They first trained the model on the entire dataset, then retrained the model on a core set and compared the difference in performance. A well-chosen core set will better represent the entire dataset, associated with a lower performance drop. It was concluded that the introduction of OOD samples to the core set does not have a significant effect on CL. However, it was not an OOD problem since the model had access to the whole dataset at the beginning of reverse CL.

Instead of narrowing our focus to specific CL strategies or adversary scenarios, we address a broader spectrum of concerns related to the security and OOD robustness in CL. That is, there is a previously unnoticed form of CF: the OODF that can severely affect CL models’ performance. Subsequent sections will detail critical properties of OODF, including its prevalence across CL strategies and settings, the challenge of its delayed detection, and the variety of how to trigger it.

## 3 Out-of-Distribution Forgetting

In CL tasks, it’s typically assumed that training and subsequent testing data are drawn from the same distribution. However, this distribution may shift, either intentionally or accidentally, as time progresses after the learning stage. It’s crucial to note that our discussion does not revolve around distribution shifts between sequential tasks (e.g., task 1 to task 2). Instead, we concentrate on the often-overlooked intra-class distribution shifts within a single task (e.g., task 1 at varying time steps). It sets our research apart from the bulk of existing studies.

In this section, we will show the influence of OODF, i.e., the catastrophic forgetting caused by the distribution shift in data between the training and inferring phases, on the artificial neural network with the mainstream CL algorithms. Firstly, we will introduce the learning paradigm and experiment procedure of

the CL task considering OODF. Next, we will evaluate OODF on various mainstream CL strategies and compare the conditions with joint learning. Finally, we will analyze the key factors that determine the extent of the influence of OODF.

### 3.1 Standard CL Paradigm

Here we take the supervised image classification problem as an example to illustrate the paradigm of CL. For experiments, we use the class incremental scenario. In the CL task, total  $K$  tasks need to be learned, and the dataset for each task is defined as

$$D^t = \{x_i^t, y_i^t\}_{i=1}^{n_t}, t = 1, 2, \dots, K \quad (1)$$

where  $t$  is the task index. The dataset for the  $t^{\text{th}}$  task has  $n_t$  pairs of labeled data. Data  $\{x_i^t\}_{i=1}^{n_t}$  and label  $\{y_i^t\}_{i=1}^{n_t}$  are sampled from distribution  $P(x^t)$  and  $P(y^t)$ , respectively. In CL, the artificial neural network  $f_{\theta_t} : X^t \rightarrow Y^t$  must learn the task once at a time. In the  $t^{\text{th}}$  task, the neural network has to optimize its parameter  $\theta_t$  according to  $D^t$ . It usually has no or very limited access to previous datasets  $D^{t-1}, D^{t-2}, \dots, D^1$ , but needs to maintain the performances on all learned classes. In the inference phase, the testing dataset in  $\{D_{test}^t\}_{t=1}^K$  is sampled from the same distribution as the training dataset.

### 3.2 OODF Paradigm

In an open and dynamic circumstance, assuming that training and testing datasets are sampled from the same distribution is not always practical. To evaluate the influence of distribution shift in data, we adjust the standard CL diagram accordingly. If a distribution shift takes place in the training dataset of the  $S^{\text{th}}$  task (i.e. intra-class shift), the data  $D^S$  will be directly replaced by the shifted data

$$\widehat{D}_{train}^S = \{\widehat{x}_i, y_i\}_{i=1}^{\widehat{n}} \cup \{x_i, y_i\}_{i=\widehat{n}+1}^n \quad (2)$$

$\widehat{D}_{train}^S$  contains  $\widehat{n}$  shifted data pairs  $\{\widehat{x}_i, y_i\}_{i=1}^{\widehat{n}}$  and  $n - \widehat{n}$  original data pairs  $\{x_i, y_i\}_{i=\widehat{n}+1}^n$ . The percentage  $r = \widehat{n}/n$  of samples  $\widehat{x}$  is used to measure the occurrence frequency (i.e. ratio) of feature shifting in the training dataset. The training procedure causing OODF is described in the Algorithm 1, reusing the mathematical notation from Sects. 3.1 and 3.2. In the referring phase, the neural network is tested on  $D_{test}^S$  sampled from the same distribution as  $D^S$ . Except for such a minor modification, the rest of the procedure in the learning is the same as that in the standard CL.

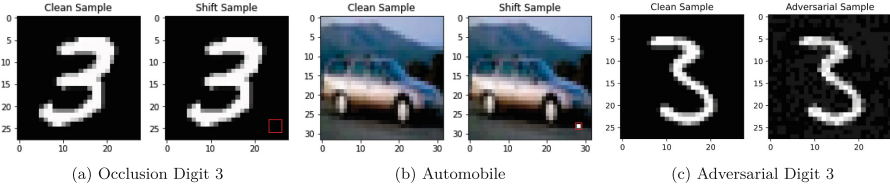
### 3.3 Introducing of the Distribution Shift

In the experiments, we constructed distribution-shifted data  $\widehat{x}$  by adding the non-shifted data  $x$  a new feature sampled from a distinct distribution  $P'(x)$ . There were no changes in the labels. In practice, we just chose a small pixel block in a fixed location in the image and set it to a constant value. The feature

position was denoted by the index  $p$ . The position of other pixels remaining the same is denoted by  $q$ , i.e.  $q = -p$ . The strength of the shifted pixels is controlled by parameter  $\epsilon$ :

$$\hat{x}[q] = x[q] \quad \hat{x}[p] = \epsilon \quad (3)$$

Figure 2a and 2b demonstrate the feature-shifting operation on two image examples from MNIST and CIFAR-10 datasets. In each panel, the left image is the original one, while the right is the corresponding image with a pixel-wise modification. The shifted pixels highlighted by the red square at the bottom right corner are vague and easily overlooked by humans. The above operation is not necessarily an attack on the CL machine, though OODF can easily be exploited for intentional sabotage. In reality, many conditions can cause such distribution shifts, e.g., slight deflection in the sensory equipment or some random, noisy perturbations. These unpredictable and hardly detectable deflections or perturbations can easily cause a feature shift in training samples.



**Fig. 2.** Distribution Shift. Red rectangle box selected the pixels that were modified in (a) and (b). Figure (c) will be discussed in later section. (Color figure online)

The distribution shift is introduced through a small pixel-wise operation causing occlusion, highlighting the significance of OODF: as the results shown

---

### Algorithm 1. Continual Learning on Distribution Shift Dataset

---

**Require:** Datasets  $\{D_{train}^t\}_{i=1}^K$ ,  $n_t$  samples in  $D_{train}^t$ , shift task-ID  $S$ , occlusion strength  $\epsilon$ , position  $p$ , percentage  $r$ , classifier with initial parameter  $f_{\theta_0}$ , loss function  $l_t(\cdot)$ , continual methods  $CL$ .

**Ensure:**  $\hat{n}_S = rn_S$

- 1:  $\hat{D}_{train}^S \leftarrow \{\hat{x}_i^S, y_i^S\}_{i=1}^{\hat{n}_S} \cup \{x_i^S, y_i^S\}_{i=\hat{n}_S+1}^{n_S}$
  - 2: **for**  $t = 1$  to  $K$ , using  $CL$  **do**
  - 3:   **if**  $t \neq S$  **then**
  - 4:     get  $\{x^t\}, \{y^t\}$  from  $D_{train}^t$
  - 5:   **else**
  - 6:     get  $\{x^t\}, \{y^t\}$  from  $\hat{D}_{train}^S$
  - 7:   **end if**
  - 8:    $\theta_t \leftarrow \arg \min_{\theta} l_t(f_{\theta}(x^t), y^t; \theta_{t-1})$
  - 9: **end for**
  - 10: **return**  $f_{\theta_K}$
-

**Table 1.** Network backbone and CL methods of experiment settings.

	Backbone	CL methods
Split MNIST-10	784-800-10	OWM [36]
	784-400-400-10	iCaRL [23],DGR [29],ER [25]
Split CIFAR-10 Split CIFAR-100	3 CNN with 3 FC	OWM,AOP [5]
	Resnet18	iCaRL,ER,DER++ [1], GDumb [20],CN-DPM [12]

in Sect. 5, even minor data augmentations can induce significant forgetting in widely tested CL methods. We also consider another form of distribution shift by using FGSM [4] to construct adversarial samples, shown in Fig. 2c. We regard shift diversity as a factor of OODF and discuss it in Sect. 6.

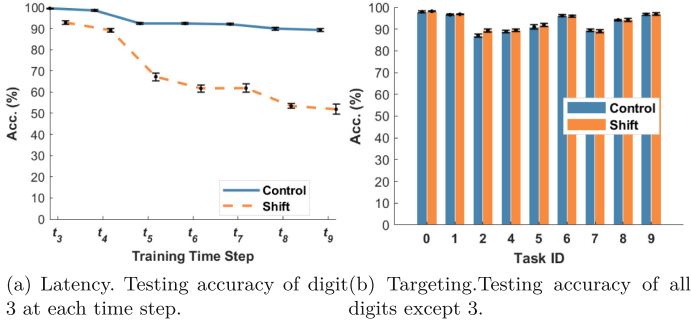
## 4 Experiment Settings

To evaluate OODF, we tested the influence of intra-class shift on all three mainstream learning strategies in classic CL tasks. The choices of the algorithm and corresponding network structure and dataset in each experiment are listed in Table 1. In all experiments, either the original code or the popular reproducing code [15] of the CL algorithms were used for evaluation. All the code had been checked in the standard CL tasks without data distribution shift. We note that we’re not aiming to evaluate the performances of different CL methods or compare performance degradation caused by distribution shifts in these methods. Instead, the purpose here is to examine the extent of OODF in CL models.

**Shift SplitMNIST-10 Task.** The MNIST dataset was divided into 10 tasks. In each task, the neural network was trained only to learn one class of handwriting digits. Each class included  $\sim 6000$  samples in the training set and  $\sim 1000$  in the testing set. The images were not pre-processed before the training. The training order of tasks was from 0, 1,  $\dots$  to 9. We took the digit 3 (task index  $S = 4$ ) as an example to illustrate the influence of the intra-class distribution shift on CL. The task choice is without specific consideration and can be replaced by other digits. The shifted training samples took the percentage  $r = 90\%$ . The shifted feature is a four-pixel square located at the bottom right corner of the image, as highlighted by the red box in Fig. 2a. The strength  $\epsilon$  was set to 64 in experiments of memory replay strategies and 32 for parameter regularization strategies.

**Shift SplitCIFAR-10&100 Task.** This task is similar to the shifted splitMNIST-10 task. The CIFAR10 dataset was divided into 10 tasks according to the category to be sequentially learned by the neural network. Each category included 6000 samples in the training set and 1000 in the testing set. The RGB images were normalized before the training. We randomly added a shifted feature of pixel square at the bottom right corner on training samples of the task  $S = 2$ . The percentage of shifted samples is  $r = 50\%$  for memory-based methods and  $r = 90\%$  for others. The square size is  $1 \times 1$  and  $2 \times 2$ , respectively. The strength is  $\epsilon = 255$  for each RGB channel. Same as above, the CIFAR100 dataset was divided into 100 tasks. To enable stochasticity, all results are collected over





**Fig. 3.** Properties of out-of-distribution forgetting.

5 independent trials, presented in the bars and tables. Details for training and distribution shifts are listed in the supplementary material.

## 5 Properties of OODF

### 5.1 Delayed Effect

As a new form of catastrophic forgetting, OODF also has a **delayed effect**. We first take the experiment of OWM on SplitMNIST-10 as an example. The experiment was conducted in a control group and a shift group. In the control group, the experiment was performed following the standard CL paradigm for comparison. In the shift group, shifted features were added to the task of  $S = 4$  and the experiment was performed following the OODF testing paradigm. The results are demonstrated in Fig. 3a. In each group, the 4<sup>th</sup> task was firstly tested immediately after the end of the current task (the time point is denoted as  $t_3$ ) and then at each time step of the experiment on the original testing dataset (the time point is denoted as  $t_i$ ). Both the control group and shift group performed well at  $t_3$ , with accuracy at  $99.54 \pm 0.16\%$  and  $92.85 \pm 0.76\%$  respectively. Although the results indicate that shift minimally affects the learning of the current task, but our primary concern is the forgetting effect it triggers during successive learning processes. As the experiment continued, the performance on the 4<sup>th</sup> task in the control group maintained high at  $89.33 \pm 0.67\%$ , indicating that the CL algorithm functioned normally and protected the previous knowledge well. As a comparison, the performance in the shift group dropped dramatically to  $51.90 \pm 2.36\%$ . The relative accuracy drop is  $10.25 \pm 0.70$  for the control group, while it is significantly worse for the shift group at  $44.11 \pm 2.42$ . We conducted similar experiments on different CL strategies, network structures, and datasets. The results are demonstrated in Tables 2, 3 and 4. In all experiments, the performance in the shift group at  $t = S$  was comparable to the control group but dropped dramatically at the end of learning  $t = K$ . These results show that distribution shifts in the data can severely degrade the function of regularization-based and remory-based CL methods, but not parameter-isolation-based methods.

**Table 2.** Out-of-distribution forgetting on MNIST. Test Acc.(%) of task  $S$  at two time steps  $t = S$  and  $t = K$ .

MNIST		Reg.		Mem.		
		OWM		iCaRL	DGR	ER
$t = S$	Control	99.54 ± 0.16		99.76 ± 0.13	99.08 ± 0.32	99.30 ± 0.39
	Shift	92.85 ± 0.76		98.67 ± 0.40	94.57 ± 1.37	97.76 ± 1.24
$t = K$	Control	89.33 ± 0.67		84.63 ± 2.54	83.40 ± 3.58	90.20 ± 1.22
	Shift	<b>51.90 ± 2.36</b>		<b>59.57 ± 6.34</b>	<b>67.43 ± 5.94</b>	<b>72.15 ± 5.97</b>

**Table 3.** Out-of-distribution forgetting on CIFAR10. Test Acc.(%) of task  $S$  at two time steps  $t = S$  and  $t = K$ .

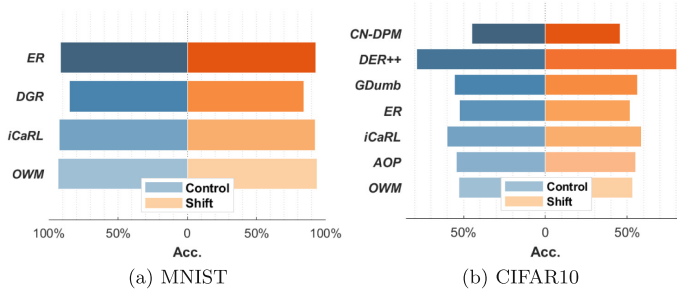
CIFAR-10		Reg.			Mem.			Iso.	
		OWM	AOP		iCaRL	ER	GDumb	DER++	CN-DPM
$t = S$	Control	94.37 ± 1.25	98.78 ± 0.27		95.52 ± 0.69	95.41 ± 1.30	96.4 ± 0.62	97.34 ± 1.03	91.15 ± 1.43
	Shift	91.30 ± 1.63	91.92 ± 1.34		92.72 ± 0.87	95.14 ± 1.37	93.18 ± 1.16	96.76 ± 1.52	89.34 ± 2.61
$t = K$	Control	52.60 ± 3.44	60.10 ± 7.10		68.88 ± 2.14	54.84 ± 6.14	74.38 ± 6.68	85.20 ± 2.27	44.76 ± 3.51
	Shift	<b>33.75 ± 4.71</b>	<b>27.70 ± 5.10</b>		<b>55.12 ± 2.74</b>	<b>46.84 ± 3.94</b>	<b>65.95 ± 7.35</b>	<b>80.82 ± 2.77</b>	<b>45.92 ± 2.06</b>

**Table 4.** Out-of-distribution forgetting on CIFAR100. Test Acc.(%) of task  $S$  at two time steps  $t = S$  and  $t = K$ . We only report the results of these three methods in the table due to compatibility issues (e.g., CN-DPM for 100 class-incremental) or intractable testing performance (e.g., Reg. based methods and DER++) for other methods.

CIFAR-100		Mem.		
		iCaRL	ER	GDumb
$t = S$	Control	94.6 ± 2.5	87.3 ± 5.3	96.2 ± 2.7
	Shift	91.8 ± 2.8	94.5 ± 1.3	94.0 ± 2.9
$t = K$	Control	47.4 ± 11.0	23.5 ± 7.6	11.4 ± 5.3
	Shift	<b>28.0 ± 12.4</b>	<b>5.5 ± 3.1</b>	<b>7.0 ± 4.6</b>

## 5.2 Targeting

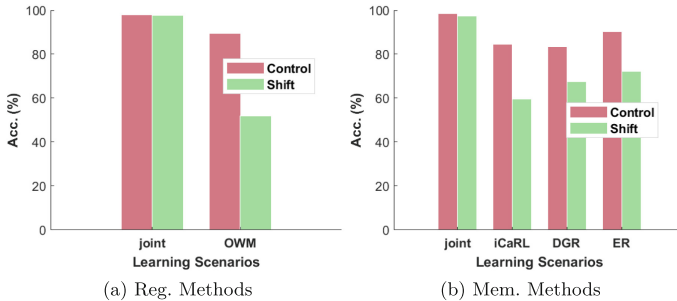
This section investigates the incidence of OODF on learned tasks. Figure 3b shows the performance of all tasks but the one recognizing digit 3 in the SplitMNIST-10 task trained with the OWM algorithm. In both the control and shift groups, the accuracy was tested at the end of the experiment. The testing accuracies of the tasks in the shift group were almost identical to those in the control group when there was no distribution shift in the training data. The result indicates that slight spillover caused by the distribution shift in a specific task affects the rest. Similarly, we further examined the rest experiments with different CL settings and algorithms.



**Fig. 4.** Comparison of non-target tasks’ accuracies between standard and shift experiments. The results were obtained by averaging the accuracies for all tasks except for task S after the whole CL learning procedure was completed. The left (right) bars for each figure are the results for the control (shift) group.

Figure 4 illustrates the average accuracy of tasks without data distribution shifting in the control and shift groups. The minor difference verifies that OODF only affects the target task with data distribution shifting in the learning sequence.

### 5.3 Continual Detrimental



**Fig. 5.** Comparison between joint learning and CL under the same distribution shift with corresponding network backbone tested on SplitMNIST-10. In each figure, the pink bar on the leftmost of each subgroup indicates training without shifts, the green bar nearby indicates learning with shifts, and the horizontal axis listed different learning strategies. (Color figure online)

Is the above phenomenon specific to CL? Or is it just a form of data poisoning working for all learning systems? To answer this question, We conducted joint learning experiments in the same setting as the above for OODF evaluation,

including the same dataset  $D_{train}$  and network structures.

$$D_{train} = \left( \bigcup_{t=1, t \neq S}^K D_{train}^t \right) \cup \hat{D}_{train}^S \quad (4)$$

In Fig. 5 we show the CL dependency of OODF by evaluating the distribution-shifted task in two different learning paradigms. We will clarify it here using splitMNIST-10 as an example. The task sequence for each CL method with different network backbones (2-layer MLP for OWM and 3-layer for iCaRL, DGR, ER as shown in Table 1) is  $D_{train}^1, D_{train}^2, \dots, \hat{D}_{train}^4, \dots, D_{train}^{10}$ . The joint learning scenario uses the union set  $D_{train} = \left( \bigcup_{t=1, t \neq 4}^{10} D_{train}^t \right) \cup \hat{D}_{train}^4$ .

Figure 5 presents the results on the testing dataset  $D_{test}^4$  (samples without distribution shift). The control and shift groups indicate the presence or absence of intra-class shifts in the training set, respectively. These results indicate that the existence of a large range of shifts which is more detrimental to CL than joint learning.

## 6 Analysis

### 6.1 Occlusion Strength

The intra-class distribution shift relies on occlusion strength  $\epsilon$ , intra-class percentage  $r$ , and the number of shifted pixels. Based on the splitMNIST-10 task and OWM algorithm in Sect. 5.1, we estimated these three factors w.r.t. test the accuracy of the target task.

**Table 5.** Impact of strength: the number of shifted pixels. OWM on MNIST.

Number	1	4	9	16
Acc.	56.34%	51.90%	44.55%	39.60%

In Fig. 6a, we evaluate the final performance of digit 3, when giving different occlusion strength levels ranging from  $\epsilon = 4$  to 128, listed on X-axis. We can see that the test accuracy dropped quickly at a low  $\epsilon$  value,  $\epsilon = 16$  for example. It indicates that even an occlusion with small strength will lead to OODF. Figure 6b shows that test accuracy stays in the plateau at a low percentage level and drops until reaching a high level, suggesting that a high percentage level is needed to cause significant OODF. We further report the results using numbers of shifted pixels as different strengths in Table 5, which shows a similar trend, indicating that the larger number of shifted pixels, the more significant of OODF.

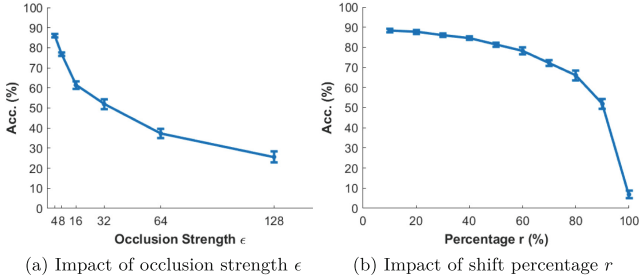


Fig. 6. Influences of distribution shift factors

## 6.2 Various Conditions of Shift

We further examined if OODF depends on specific types of distribution shifts. To this end, we replaced the explicit distribution shift (i.e. occlusion) in Sect. 3.3 with an implicit one, i.e., adversarial samples (Fig. 2c), and kept other settings the same. The results show a trend consistent with the occlusion condition. We tested digit 3 at  $t_3$  and  $t_9$ , it dropped from  $94.28 \pm 0.62\%$  to  $22.78 \pm 2.47\%$ , compared to occlusion,  $92.85 \pm 0.76\%$  to  $51.90 \pm 2.36\%$ .

## 6.3 Shift Position in the Learning Sequence

Shift position used in results above is in the beginning of the learning sequence, e.g. 4<sup>th</sup> of 10 in splitMNIST-10. We evaluated whether the position matters for OODF. We have conducted the experiments on CIFAR100 by iCaRL (Table 6), with shift task position varying (in the middle or the tail of the sequence, task  $S = 50$  or  $90$ , as  $S = 2$  was already shown in maintext). The results of the relative accuracy drop in two groups indicate that while task location can cause different levels of degradation due to the original CF, the OODF effect can still induce additional forgetting based on CF.

Table 6. Effect of different shift position on OODF. Experiments on CIFAR100,  $K = 100$  (total number of tasks),  $S = 2, 50, 90$  (position of the shift class, starting from 1). All results are collected over 5 independent trials.

CIFAR-100		$S = 2$	$S = 50$	$S = 90$
Acc. of task $S$ at $t = S$ (%)	Control	$94.6 \pm 2.5$	$53.0 \pm 3.1$	$36.6 \pm 5.8$
	Shift	$91.8 \pm 2.8$	$53.2 \pm 6.3$	$34.6 \pm 1.7$
Acc. of task $S$ at $t = K$ (%)	Control	$47.4 \pm 11.0$	$31.8 \pm 5.5$	$32.4 \pm 3.1$
	Shift	$28.0 \pm 12.4$	$28.4 \pm 3.4$	$29.4 \pm 4.3$
relative Acc. drop (%)	Control	$50.0 \pm 10.9$	$40.1 \pm 8.9$	$10.7 \pm 7.1$
	Shift	<b><math>69.2 \pm 14.3</math></b>	<b><math>46.2 \pm 7.2</math></b>	<b><math>15.2 \pm 9.9</math></b>

## 6.4 Different Percentage $r$ and Strength $\epsilon$

Our aim is not to compare the vulnerability of different methods to OODF under identical conditions. Taking the performance of OWM on the splitMNIST-10 task as an example: when  $\epsilon = 64$ , the performance is  $37.3 \pm 2.4$ , significantly lower than the control group  $89.33 \pm 0.67$  (Table 2 and Fig. 6a). We focus on the performance decline trends relative to each method’s own control group, rather than specific differences between methods. Our objective is to demonstrate the existence of intra-class distribution shifts that can influence CL algorithms to produce OODF. Notably, these shifts do not significantly impact joint learning scenarios, even at maximum percentage  $r = 90\%$  and occlusion strength  $\epsilon = 64$  (Fig. 5).

## 6.5 Mechanism of OODF

We hypothesized that frequently occurring shifts can serve as informative features for classification. This compromises the mechanism designed to protect the intrinsic features for the learned class, leading to severe CF in subsequent learning, especially when the less-protected features overlap with the features in new classes.

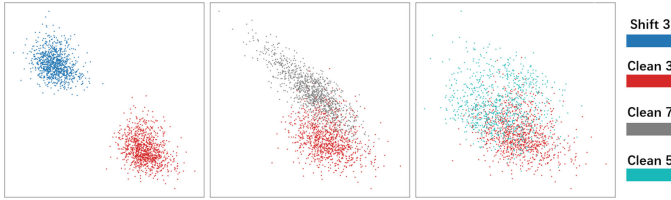
**Table 7.** Test accuracy of the digit 3 after learning each task.

Task-ID	3	4	5	6	7	8	9
Control	99.54%	98.58%	92.40%	92.40%	92.05%	90.02%	89.34%
Shift	94.28%	88.62%	<b>50.64%</b>	41.21%	37.91%	<b>24.80%</b>	22.78%

Specifically, we take the results in Sect. 6.2 for analysis. The in-process test accuracy of digit 3 from  $t_0$  to  $t_9$  was shown in Table 7, and the performance drops significantly at  $t_5$  and  $t_8$ . Let  $\mathcal{D}_3$  be the distribution of clean digit 3 and  $\hat{\mathcal{D}}_3$  be the distribution of shifted digit 3. Assume  $\mathcal{D}'_3 = \mathcal{D}_3 n(\mathcal{D}_3 \cap \hat{\mathcal{D}}_3)$  and  $\hat{\mathcal{D}}'_3 = \hat{\mathcal{D}}_3 n(\mathcal{D}_3 \cap \hat{\mathcal{D}}_3)$ . We make the following conjecture: (i) In the learning process of the OODF scenario, the feature of  $\mathcal{D}_3 \cap \hat{\mathcal{D}}_3$  was protected. Meanwhile, features of  $\hat{\mathcal{D}}_3$  overlap with that of subsequent tasks. (ii) Performance on clean 3 mainly depends on  $\mathcal{D}'_3$  rather than  $(\mathcal{D}_3 \cap \hat{\mathcal{D}}_3)$ . Taken together, OODF happens on digit 3. We show that the accuracy of digit 3 drops significantly after subsequently learning 5 and 8.

To test the hypothesis, we constructed a 3-layer binary classification MLP that distinguishes  $\mathcal{D}_3$  and  $\hat{\mathcal{D}}_3$  as large as possible (Fig. 7. row 1, column 1). Input any other digit through this MLP, we take  $\mathbb{R}^2$  output vector and construct a feature map. Consistent with the hypothesis, we found 5 and 8 overlap more with clean 3 than other digits (e.g. digit 7. row 1, column 2).

These results revealed why parameter-isolation-based methods are not sensitive to OODF (Table 3). The feature space representations in these methods



**Fig. 7.** Feature maps of digits.

vary from task to task, and more importantly, they are independent from each other. In contrast, regularization-based and memory-based methods share a public representation space for every task, which causes interference. Despite the robustness of the parameter-isolation-based methods towards OODF, they may be not suited to deal with a large amount of CL tasks due to the increasing structural complexity. Our results thus highlight the need to develop regularization-based and memory-based approaches that are more robust to OODF.

**Table 8.** Additional rejection category for alleviating OODF. Experiments on MNIST,  $K = 10$  (total number of tasks),  $S = 2$  (starting from 1)

MNIST		OWM (w/o rej.)	OWM (w/ rej.)
Acc. of task $S$ at $t = S$ (%)	Control	$99.54 \pm 0.16$	$99.36 \pm 0.26$
	Shift	$92.85 \pm 0.76$	$99.35 \pm 0.21$
Acc. of task $S$ at $t = K$ (%)	Control	$89.33 \pm 0.67$	$86.31 \pm 1.24$
	Shift	<b><math>51.90 \pm 2.36</math></b>	<b><math>85.17 \pm 1.04</math></b>
relative Acc. drop (%)	Control	$10.25 \pm 0.70$	$13.13 \pm 1.38$
	Shift	<b><math>44.11 \pm 2.42</math></b>	<b><math>14.27 \pm 1.13</math></b>

## 6.6 Proposal for Improving OODF

Based on the mechanism discussed in Sect. 6.5, reducing task-independent subspace in the feature space may help prevent OODF. We propose introducing a rejection category to separate the classifier from CL methods. Empirically, this approach is effective in mitigating OODF. Specifically, we add an additional neuron at the output layer, enabling an 11-way classification. Samples like Gaussian noise are set to be the 11<sup>th</sup> class. We have finished the experiments on splitMNIST-10 with OWM. Samples 4x the size of the MNIST dataset (60k samples in MNIST training set) are generated as an independent task, which is inserted to the beginning of the learning sequence. OWM with rejection category (OWM w/ rej.) significantly alleviates OODF, exhibiting in the relative accuracy drop  $14.27 \pm 1.13$  is much lower than the OWM without rejection

$44.11 \pm 2.42$  (OWM w/o rej.), in Table 8. While our primary focus in this work is to highlight the importance of OODF, we acknowledge that solving this problem is crucial. The approach presented here is a preliminary attempt, which will be thoroughly investigated in future.

## 7 Conclusion

In this work, we identify a new phenomenon of catastrophic forgetting, named out-of-distribution forgetting, and demonstrate how it can significantly affect the robustness of CL. Although OODF is described here in the image classification task under the class incremental scenario, it is straightforward to extend to other CL tasks in computer vision or natural language processing. OODF reveals the vulnerability of current CL methods in dealing with intra-class distribution shifts, which could be introduced intentionally or by unnoticed perturbations. This is well-conceivable in both attacking or accidental scenarios.

More generally, our work suggests that the catastrophic forgetting problem in CL is more complex than we previously recognized, and it is likely that other forms of CF cannot be dealt with by the majority of current CL approaches may exist. Thus, it is of theoretical and practical importance to investigate the issue of CF more comprehensively, which will guide the development of more robust CL approaches that can work in complex environments.

## References

1. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 15920–15930 (2020)
2. Caccia, M.: Online fast adaptation and knowledge accumulation (OSAKA): a new approach to continual learning. *Adv. Neural. Inf. Process. Syst.* **33**, 16532–16545 (2020)
3. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2015)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2015)
5. Guo, Y., Hu, W., Zhao, D., Liu, B.: Adaptive orthogonal projection for batch and online continual learning. *Proc. AAAI Conf. Artif. Intell.* **36**, 6783–6791 (2022)
6. Guo, Y., Liu, M., Li, Y., Wang, L., Yang, T., Rosing, T.: Attacking Lifelong Learning Models with Gradient Reversion (2019). <https://openreview.net/pdf?id=SJlpy64tvB>
7. Hoge, E., Popescu, A., Onchis, D., Petit, G.: FeTrIL++: feature translation for exemplar-free class-incremental learning with hill-climbing (2024)
8. Hsu, Y.C., Liu, Y.C., Ramasamy, A., Kira, Z.: Re-evaluating continual learning scenarios: a categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488* (2019)



9. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 19–35 (2018)
10. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *PNAS* **114**(13), 3521–3526 (2017)
11. Kudithipudi, D., et al.: Biological underpinnings for lifelong learning machines. *Nat. Mach. Intell.* **4**(3), 196–210 (2022)
12. Lee, S., Ha, J., Zhang, D., Kim, G.: A neural Dirichlet process mixture model for task-free continual learning. arXiv preprint [arXiv:2001.00689](https://arxiv.org/abs/2001.00689) (2020)
13. Li, H., Ditzler, G.: Targeted data poisoning attacks against continual learning neural networks. In: 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2022)
14. Li, Y., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: a survey. *IEEE Trans. Neural Netw. Learn. Syst.*, 1–18 (2022)
15. Mai, Z., Li, R., Jeong, J., Quispe, D., Kim, H., Sanner, S.: Online continual learning in image classification: an empirical survey. *Neurocomputing* **469**, 28–51 (2022)
16. Marsocci, V., Scardapane, S.: Continual Barlow Twins: continual self-supervised learning for remote sensing semantic segmentation. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **16**, 5049–5060 (2023)
17. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: the sequential learning problem. In: Bower, G.H. (ed.) *Psychology of Learning and Motivation*, vol. 24, pp. 109–165. Academic Press (1989)
18. Mundt, M., Hong, Y., Pliushch, I., Ramesh, V.: A wholistic view of continual learning with deep neural networks: forgotten lessons and the bridge to active and open world learning. *Neural Netw.* (2023)
19. Mundt, M., Pliushch, I., Majumder, S., Hong, Y., Ramesh, V.: Unified probabilistic deep continual learning through generative replay and open set recognition. *J. Imaging* **8**(4), 93 (2022)
20. Prabhu, A., Torr, P.H.S., Dokania, P.K.: GDumb: a simple approach that questions our progress in continual learning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) *Computer Vision - ECCV 2020*, pp. 524–540 (2020)
21. Ratcliff, R.: Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychol. Rev.* **97**, 285–308 (1990)
22. Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., Almahairi, A.: Progressive Prompts: continual learning for language models (2023)
23. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: incremental classifier and representation learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010 (2017)
24. Ren, K., Zheng, T., Qin, Z., Liu, X.: Adversarial attacks and defenses in deep learning. *Engineering* **6**(3), 346–360 (2020)
25. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. In: *Advances in Neural Information Processing Systems*, vol. 32 (2019)
26. Rusu, A.A., et al.: Progressive neural networks (2016)
27. Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M.H., Sabokrou, M.: A Unified survey on anomaly, novelty, open-set, and out-of-distribution detection: solutions and future challenges. *Trans. Mach. Learn. Res.* (2022)
28. Shen, Z., et al.: Towards out-of-distribution generalization: a survey. arXiv preprint [arXiv:2108.13624](https://arxiv.org/abs/2108.13624) (2021)
29. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. arXiv preprint [arXiv:1705.08690](https://arxiv.org/abs/1705.08690) (2017)

30. Umer, M., Dawson, G., Polikar, R.: Targeted forgetting and false memory formation in continual learners through adversarial backdoor attacks. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2020)
31. Umer, M., Polikar, R.: Adversarial targeted forgetting in regularization and generative based continual learning models. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2021)
32. Umer, M., Polikar, R.: False memory formation in continual learners through imperceptible backdoor trigger. arXiv preprint [arXiv:2202.04479](https://arxiv.org/abs/2202.04479) (2022)
33. van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. arXiv preprint [arXiv:1904.07734](https://arxiv.org/abs/1904.07734) (2019)
34. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–20 (2024)
35. Yang, B., et al.: Continual object detection via prototypical task correlation guided gating mechanism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9255–9264 (2022)
36. Zeng, G., Chen, Y., Cui, B., Yu, S.: Continual learning of context-dependent processing in neural networks. *Nat. Mach. Intell.* **1**(8), 364–372 (2019)
37. Zhang, G., Wang, L., Kang, G., Chen, L., Wei, Y.: SLCA: slow learner with classifier alignment for continual learning on a pre-trained model. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 19148–19158 (2023)



# Generating Multi-objective Fronts from Streamed Data Using Nested List

Arnabi Mukherjee<sup>1</sup>, Sourab Mandal<sup>1</sup>, and Paramartha Dutta<sup>1</sup>

Department of Computer and System Sciences, Visva-Bharati,  
Santiniketan, WB, India  
smvb.rs@gmail.com

**Abstract.** In the sphere of managing multiple, conflicting objectives concurrently, non-dominated sorting emerges as a pivotal method guiding decision-making towards optimal solutions by generating one or more Fronts. While numerous algorithms exist for multi-objective non-dominated sorting on static data points, there remains a scarcity of the same on streamed or online data points. This study focuses on the critical realm of handling real-time or online data streams to craft an algorithm specifically tailored to manage such real-time and critical data scenarios. Furthermore, this research not only introduces a novel algorithm that utilizes a simple yet effective nested list structure mechanism to perform the task of non-dominance sorting for streamed data but also evaluates its performance by checking its correctness with the existing Fast Non-dominated Sorting algorithm which is used in both Non-dominated Sorting Genetic Algorithm (NSGA-III) and Non-dominated Sorting Genetic Algorithm II (NSGA-II). The efficacy of the algorithm is also proven by showing its applicability on numerous benchmark datasets. The proposed mechanism shows complexity  $O(MN)$  in terms of space, whereas  $O(MN)$  is the time complexity in the best-case scenario, and the worst-case as well as average-case complexity for the same is  $O(MN^2)$ . Here,  $M$  denotes the number of objective functions and  $N$  indicates the population size.

**Keywords:** NSGA-III · Fast Non-dominated Sorting · nested list structure · NSGA-II · streamed data · online non-dominated sorting · non-dominated sorting for streamed data

## 1 Introduction

In numerous real-world decision-making scenarios, individuals and organizations often encounter situations where multiple, often conflicting, objectives need to be considered simultaneously. Whether in engineering, finance, logistics, or various other domains, the pursuit of optimal solutions frequently involves balancing multiple criteria. Traditional single-objective optimization approaches,

---

A. Mukherjee and S. Mandal—Equal contributions to this work.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15309, pp. 128–143, 2025.  
[https://doi.org/10.1007/978-3-031-78189-6\\_9](https://doi.org/10.1007/978-3-031-78189-6_9)

while effective in addressing individual goals, often fall short of capturing the complexities inherent in these multifaceted problems. Multi-objective optimization (MOO) plays an important role in scenarios like these. NSGA-II and NSGA-III are among the methodologies frequently applied in domains needing MOO. Both are two-pass algorithms, commencing with the Fast Non-dominated Sorting (FNDS) step, which categorizes solutions into distinct Fronts according to their dominance relationships. Despite its simplicity and effectiveness, FNDS faces criticism for its considerable time and space complexities. Furthermore, existing algorithms, including FNDS, solely operate with static data points. In our work, we aim to introduce an alternative online mechanism to address these limitations.

## 1.1 Prerequisites

Requisite terminologies, definitions, and concepts are presented here.

**Definition 1 (Multi-Objective Optimization Problem).** *In mathematical terms, the idea of MOO can be articulated through the following formulation:*

$$\begin{aligned} \text{Maximize: } & [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})], \\ \text{Subject to: } & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, k, \\ & h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, p. \end{aligned}$$

Here,  $N$  represents the number of decision variables, denoted as a vector  $\mathbf{x} = [x_1, x_2, \dots, x_N]$ . These variables must satisfy a set of  $k$  inequality constraints and  $p$  equality constraints. Simultaneously, the objective is to maximize a vector of  $M$  objective functions, where each element corresponds to a distinct objective function. These objective functions act as mathematical representations of diverse, and potentially conflicting, performance criteria. Consequently, the concept of “optimization” in this context involves discovering a solution that offers acceptable values for all objective functions, as determined by the decision maker [3].<sup>1</sup>

**Definition 2 (Decision Variable).** *The decision variables in an optimization problem are numerical values chosen for the purpose of optimization. These values are represented as  $x_i$ , where  $i = 1, 2, \dots, N$ . The vector  $\mathbf{x}$ , consisting of  $N$  decision variables, is expressed as follows:  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$  [3].*

**Definition 3 (Objective Function).** *Objective functions are mathematical expressions that evaluate the performance of a solution based on the values of decision variables. These functions are essential for assessing the quality of a solution and can be mathematically represented as  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})]^T$ . In this representation,  $M$  denotes the number of objective functions, and for MOO Problems (MOOPs), it holds that  $M \geq 2$  [5].*

<sup>1</sup> The terms objective space data points and elements have been used interchangeably.

**Definition 4 (Dominance Relation).** *A solution within the decision space, represented by the vector  $S_1$  that includes decision variables, is deemed to dominate another solution  $S_2$  (indicated as “ $S_1 \succ S_2$ ”) when it fulfils these two conditions: (1)  $S_1$  is not worse than  $S_2$  for any objective function. (2)  $S_1$  outperforms  $S_2$  for at least one objective function [3].*

**Definition 5 (Non-dominance Relation).** *Two arbitrary solutions, denoted as  $S_1$  and  $S_2$  within a decision space, are considered to be in a non-dominance or indifference relation, symbolized by “ $S_1 \sim S_2$ ,” if neither  $S_1$  dominates  $S_2$  (expressed as  $S_1 \not\succeq S_2$ ) nor does  $S_2$  dominate  $S_1$  (indicated by  $S_2 \not\succeq S_1$ ) [5]. In this text, non-dominance and indifference are used interchangeably.*

**Definition 6 (Pareto Optimal Set).** *Pareto optimality is achieved by a solution when no other solution can improve one objective without worsening another. The set containing all Pareto optimal solutions is called the “Pareto optimal set,” and the corresponding points in the objective space form the “Pareto Front” [5].*

## 1.2 Literature Survey

Numerous established algorithms are available for conducting non-dominated sorting. Table 1 showcases several significant algorithms designed for this purpose. It is worth mentioning that the TNS method introduced by Jensen is applicable solely to scenarios with two objective functions. Likewise, the ENLU technique proposed by Li et al. accounts for trivial cases when assessing the best-case scenario.

## 1.3 Gap Identification and Motivation

In Subsect. 1.2, diverse non-dominated sorting approaches were explored, each with unique methodologies. However, existing algorithms primarily focus on static data, neglecting strategies to tackle the streamed data scenarios. Static data remains constant once captured, serving as a stable reference point acquired at defined timestamps. Streamed data, on the contrary, continuously transmits dynamic information in real-time without having any predefined endpoint, necessitating immediate processing for instant analysis. We suggest a straightforward yet effective mechanism using a nested list structure to address challenges posed by streamed data. It is important to note that, while existing state-of-the-art algorithms may perform efficiently on static data, none currently address streamed data. The ability to work with streamed data is crucial due to its relevance in real-life applications. We evaluated its performance on benchmark datasets to demonstrate its proficiency in analyzing such data.

## 1.4 Salient Points of the Proposed Approach

The proposed mechanism, which utilizes a nested linked list structure for performing the non-dominated sorting on the MOO framework, shows the following salient points.

**Table 1.** Various non-dominated sorting algorithms and their complexities.

Approach	Year	Time Complexity		Space
		Best-case	Worst-case	Complexity
NDS by Srinivas et al. [21]	1994	$O(MN^2)$	$O(MN^3)$	$O(N)$
FNDS by Deb et al. [4]	2002	$O(MN^2)$	$O(MN^2)$	$O(N^2)$
TNS by Jensen [12]	2003	$O(N \log^{M-1} N)$	$O(N \log^{M-1} N)$	$O(MN)$ [18]
ND rank Sort by Deb et al. [7]	2005	$O(MN^2)$	$O(MN^2)$	$O(N)$
D&C approach by Fang et al. [9]	2008	$O(MN \log N)$	$O(MN^2)$	$O(MN)$ [18]
Principle of Arena by Tang et al. [22]	2008	$O(MN\sqrt{N})$	$O(MN^2)$	Unspecified
Deductive Sort by McClymont et al. [17]	2012	$O(MN\sqrt{N})$	$O(MN^2)$	$O(N)$
Corner Sort by Wang et al. [23]	2013	$O(MN\sqrt{N})$	$O(MN^2)$	$O(N)$
Generalized TNS by Fortin et al. [10]	2013	$O(N \log^{M-1} N)$	$O(MN^2)$	$O(MN)$
ENS-SS by Zhang et al. [25]	2015	$O(MN\sqrt{N})$	$O(MN^2)$	$O(1)$
ENS-BS by Zhang et al. [25]	2015	$O(MN \log N)$	$O(MN^2)$	$O(1)$
M-Front by Drozdik et al. [8]	2015	$O(MN)$	$O(MN^2)$	Unspecified
BOS by Roy et al. [20]	2016	$O(MN \log N)$	$O(MN^2)$	$O(MN)$
AENS by Zhang et al. [27]	2016	$O(N\sqrt{N})$	$O(N^2)$	Unspecified
ENLU by Li et al. [13]	2016	$O(M)$	$O(MN^2)$	Unspecified
HNDS by Bao et al. [1]	2017	$O(MN\sqrt{N})$	$O(MN^2)$	$O(N)$
TENS by Zhang et al. [26]	2017	$O(MN \log N / \log M)$	$O(MN^2)$ [18]	$O(N)$
DDA-NS by Zhou et al. [29]	2017	$O(MN \log N)$	$O(MN^2)$	$O(N^2)$
ENS-NDT by Gustavsson et al. [11]	2018	$MN \log N$ , if $M > \log N$ $N \log^2 N$ , otherwise	$O(MN^2)$	$O(N \log N)$
MN-DS by Moreno et al. [19]	2018	$O(N \log N)$	$O(MN^2)$	$O(N^2)$
D&C approach by Mishra et al. [18]	2019	$O(N \log N + MN)$	$O(MN^2)$	$O(N)$
SETNDS by Xue et al. [24]	2020	$O(MN \log N)$	$O(MN^2)$	$O(MN)$
RO by Burlacu et al. [2]	2022	$O(MN \log N)$	$O(MN^2)$	$O(N)$
RS by Burlacu et al. [2]	2022	$O(MN \log N)$	$O(MN^2)$	$O(N^2)$
PNDS by Mandal et al. [15]	2023	$O(MN^2)$	$O(MN^2)$	Unspecified
<b>Proposed mechanism</b>	2024	$O(MN)$ , for a large $N$ $O(M)$ , if $N = 2$ $O(1)$ , if $N = 1$	$O(MN^2)$	$O(MN)$

1. It accurately and comprehensively generates Fronts for any number of objective functions and any values of the objective space points.
2. Additionally, it adeptly handles streamed data.
3. Despite processing streamed data in a sequential manner, the mechanism remains unaffected in its ultimate generation of Fronts. In other words, the mechanism is independent of the order in which data points arrive.
4.  $O(MN)$  is the best-case time complexity for the mechanism, which is at par with the best available time complexity in the literature for the best-case scenario, while  $O(MN^2)$  is the time complexity in the case of the worst-case scenario, which is not worse than the existing algorithms.
5.  $O(MN)$  is the space complexity for the nested list structure mechanism.

### 1.5 Organization of the Paper

The paper is organized into four main sections. Introduction, Sect. 1, covers preliminary terminologies, a literature review, subsection on identifying gaps and motivation, and salient points of the present endeavour. Section 2 outlines the algorithm of the proposed methodology, its operational procedure, and specifications of the benchmark datasets used. In Sect. 3, the results derived from the suggested approach for different benchmark datasets along with the time and space complexity analyses of the mechanism are demonstrated. Here, the correctness and completeness of the method are also presented. Finally, Sect. 4 offers the conclusion of the present endeavour.

## 2 A Nested List Structure for Non-dominated Sorting of Streamed Data Elements

In this section, a comprehensive algorithm for the non-dominated sorting of streamed data points is presented. The algorithm's functioning is thoroughly discussed, employing a detailed case diagram to illustrate its operation. Additionally, information regarding the datasets employed to validate the proposed methodology is also provided.

**Algorithms of the Proposed Approach and Their Description.** The steps outlined in Algorithm 2 detail the comprehensive procedure employed to execute non-dominated sorting for streamed data utilizing a nested linked list structure. This process is facilitated with the assistance of the sub-function CHECK as presented in Algorithm 1. Algorithm 1 determines whether one objective point dominates another, as per Definition 4. Mathematically, consider the vector, which is nothing but an objective space point that is not yet placed in any front, “incoming\_element =  $(a_1, a_2, \dots, a_M)$ ,” where  $M$  represents the number of objective functions. Similarly, let “current\_Front =  $(b_1, b_2, \dots, b_M)$ ” be another vector of the same size. The “incoming\_element” dominates “current\_Front” (denoted as “incoming\_element  $\succ$  current\_Front”) if  $\forall i, 1 \leq i \leq M, a_i \geq b_i$ , and  $\exists i, 1 \leq i \leq M, a_i > b_i$ . Without loss of generality, all objective functions are assumed to be maximized simultaneously. To illustrate the concept of dominance

with a simple example: let  $\text{incoming\_element} = (2, 4, 6, 8)$  and  $\text{current\_Front} = (1, 3, 5, 8)$ , where  $M = 4$ . It is clear that  $\text{incoming\_element} \succ \text{current\_Front}$  because, for every objective function, the values of  $\text{incoming\_element}$  are greater than or equal to those of  $\text{current\_Front}$ , and there is at least one objective function where the value of  $\text{incoming\_element}$  is better than that of  $\text{current\_Front}$ . In another example,  $\text{current\_Front} \succ \text{incoming\_element}$ , where  $\text{current\_Front} = (20, 10, 80, 60)$ , and  $\text{incoming\_element} = (18, 8, 6, 40)$ . Similarly, two vectors or objective space points are considered to be in an indifferent or non-dominance relationship if neither vector dominates the other. A simple example can illustrate this concept. Let  $\text{incoming\_element} = (2, 4, 6, 8)$  and  $\text{current\_Front} = (1, 5, 7, 10)$ . In this case, neither vector meets the conditions required to dominate the other, so they maintain indifferent or non-dominance relationship. So, there are three possible outcomes for the function CHECK: either  $\text{incoming\_element} \succ \text{current\_Front}$ , or  $\text{current\_Front} \succ \text{incoming\_element}$ , or both vectors are in an indifferent relationship. Algorithm 2 demonstrates the entire process of the proposed mechanism. This mechanism processes each objective space point sequentially, equipping it to handle streamed data. Until the incoming element finds its appropriate Front, all other points are temporarily held in a queue or file structure. Algorithm 3 serves as the driver algorithm for Algorithm 2. Initially, the first point creates its own Front if no other points or Fronts exist, as depicted in steps 4 to 6 of Algorithm 2 and shown in Fig. 1 as *Case – 1*. If at least one Front exists, the incoming element checks its dominance relation with all existing Front elements, starting from Front 1 (the best Front in the structure) to Front K (the worst Front, where  $K > 1$ , at the end of the nested linked structure). This process occurs within the while loop in step 7 of Algorithm 2. The CHECK function, used for dominance checks, can yield four following cases. Figure 1 illustrates *Case – 2*, where the incoming element is indifferent to all elements in the existing Fronts and thus joins that Front, avoiding further searches for its position. In Algorithm 2, the variable “*num\_check*” facilitates this task. When the CHECK function returns 1 (*decider* = 1) for each comparison between the *incoming\_element* and the current Front elements, “*num\_check*” increments by one. If “*num\_check*” equals the number of elements in the current Front, indicating indifference with all existing elements, the incoming element is placed in that Front (steps 8 to 12 and 23 to 26 of Algorithm 2). If the value of the variable *decider* equals 2, the *incoming\_element* is dominated by at least one current Front element, preventing it from joining that Front. It then checks the next Front and continues recursively until it either finds the appropriate position or a new Front is created at the end for the element, as depicted in Fig. 1 as *Case – 5* (steps 13 to 17 and 40 to 44 of Algorithm 2). Steps 18 to 21 and 27 to 33 of Algorithm 2 represent *Case – 4* in Fig. 1, where the *incoming\_element* dominates some current Front elements. These dominated elements are removed from the Front and placed in a queue named “*homeless*.” After the incoming element’s relation with all current Front elements is assessed, it joins the current Front, and the elements in “*homeless*” are reassessed for their respective Front positions before processing



subsequent incoming elements. Finally, if the *incoming\_element* dominates all current Front elements, a new Front is created before the current Front, and the incoming element is inserted into the new Front, updating the rest of the Front numbers accordingly. This scenario is depicted in Fig. 1 as *Case* – 3. Steps 18 to 21 and 34 to 37 of Algorithm 2 illustrate this using the variable “*add\_self*.” It is important to note that *Cases* – 1 through *Cases* – 5 cover all possible scenarios for this mechanism. The implemented code along with the dataset for the same is provided in the footnote as a link.<sup>2</sup>

---

### Algorithm 1. Dominance Check

---

```

1: function CHECK(incoming_element, current_Front_element)
2:   less_than_current_Front ←
       all( $x < y$  for  $x, y$  in zip(incoming_element, current_Front_element))
3:   greater_than_current_Front ←
       all( $x > y$  for  $x, y$  in zip(incoming_element, current_Front_element))
4:   if less_than_current_Front then
5:     return 2    ▷ “incoming_element is dominated by current_Front_element”
6:   else if greater_than_current_Front then
7:     return 0    ▷ “incoming_element dominates current_Front_element”
8:   else
9:     return 1    ▷ “incoming_element shows indifference with
       current_Front_element”
10:  end if
11: end function

```

---

## 2.1 Benchmark Data-Set Specification

Below are the utilized benchmark datasets along with their particulars.

1. **Zitzler-Deb-Thiele-1 (ZDT-1) Dataset:** ZDT-1 requires two objective functions to be minimized which is translated to maximization problem for the convenience of the utilization of the proposed mechanism. For the validation of the mechanism, two different samples are taken of size 500 solutions and 1000 solutions. The details of the dataset can be found here [15, 30].
2. **Leading Ones and Trailing Zeros (LOTZ) and OneMinMax Datasets:** LOTZ and OneMinMax benchmark are two-objective maximization type optimization functions. The sample size for the OneMinMax benchmark is taken as 128 whereas the LOTZ dataset features a sample size of 256 data points. The details of the benchmark functions can be found here [15, 28].
3. **A Real Life Application Oriented Problem of Vehicle Crashworthiness:** This benchmark, based on real-life applications, focuses on optimizing

---

<sup>2</sup> [Click here to find the implemented code.](#)

---

**Algorithm 2.** A nested list structure (NLS) based non-dominated sorting algorithm for streamed data elements in a MOO framework

---

**Input:** Objective space point “*incoming\_element*” one at a time from a stream or a file.

**Output:** Non-overlapping Fronts

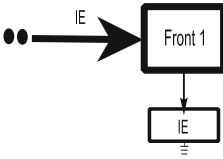
```

1: function NLS(incoming_element)
2:   count_front = 0
3:   add_self = 0 ▷ Holds the information whether there are any elements in the queue or
   not
4:   if fronts == 0 then                                ▷ “fronts” holds the number of fronts in any instant
5:     INSERT a new front and add incoming_element in the front;
6:   end if
7:   while fronts do
8:     num_check = 0                                     ▷ Holds the information for non-dominance relation of two
   elements
9:     for  $i \leftarrow 0$  to length(current_front) do
10:      decider = CHECK(incoming_element, current_front[ $i$ ])
11:      if decider == 1 then                             ▷ If it shows indifference for all the current front
   elements
12:        num_check = num_check + 1
13:      else if decider == 2 then                         ▷ If it gets dominated by any of the current front
   elements
14:        if no next Front is found then
15:          INSERT a new Front and ADD incoming_element to the Front.
16:        end if
17:        count_front + = 1                               ▷ Go to the next Front
18:      else ▷ When the incoming element dominates some or all the elements of the
   current Front
19:        add_self + = 1
20:        ADD the current dominated elements in a queue “homeless”
21:      end if
22:    end for
23:    if num_check == length(current_front) then
24:      ADD the incoming_element to the current_front
25:      return
26:    end if
27:    if add_self < length(current_front) then
28:      DELETE the dominated elements from the current_front
29:      while homeless ≠ EMPTY do
30:        element = dequeue(homeless)
31:        NLS(element)
32:      return
33:    end while
34:    else if add_self == length(current_front) then
35:      INSERT a new front before current_front
36:      ADD incoming_element in the new front
37:    end if
38:    fronts ← fronts.next                               ▷ Go to the next Front. Loop controller updation.
39:  end while
40:  if count_front = length(fronts) then
41:    ▷ If the incoming_element doesn't fit even in the last existing Front
42:    CREATE a new Front at the end of all the fronts
43:    ADD incoming_element to this front
44:  end if
45: end function

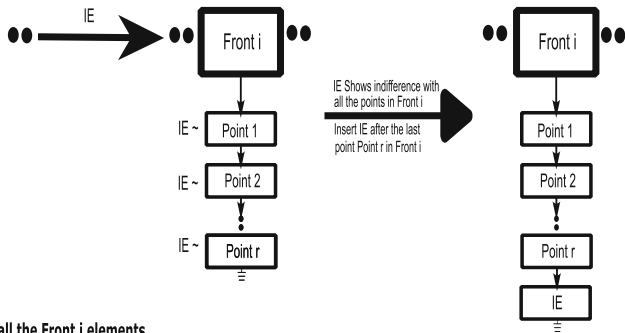
```

---

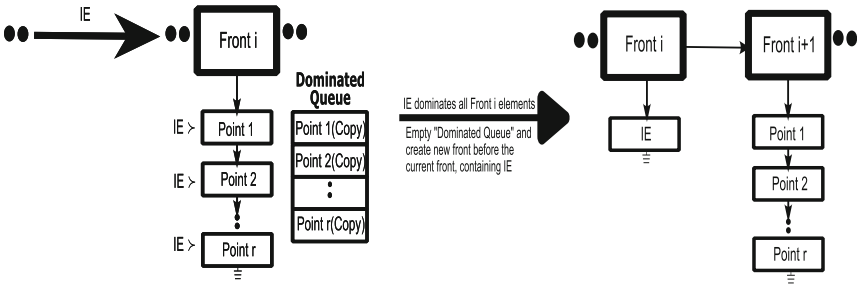
Case - 1: First entry of the incoming element (IE)



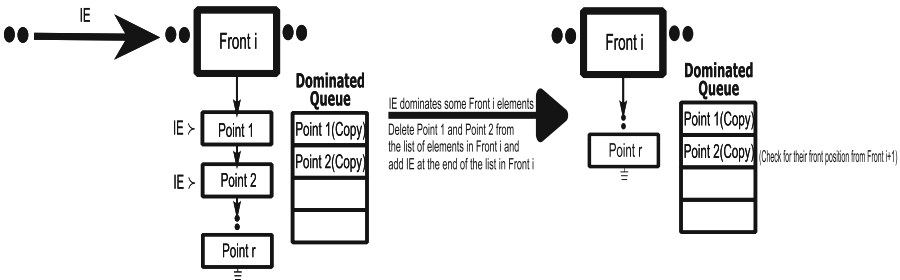
Case - 2: Incoming element (IE) shows indifference with all the Front i elements



Case - 3: Incoming element (IE) dominates all the Front i elements



Case - 4: Incoming element (IE) dominates some Front i elements



Case - 5: Incoming element (IE) is dominated by some Front i element

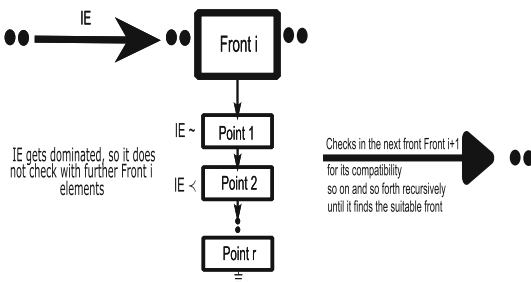


Fig. 1. Process and different cases for the proposed approach using a nested list structure to handle stream data.

**Algorithm 3.** Driver algorithm for NLS

---

```

1:  $dSize = 0$ 
2: while True —  $dSize == N$  do                                ▷  $N$ : Size of the population
3:   NLS( $incoming\_element$ )                                ▷  $incoming\_element$ : Objective space points
4:    $dSize+ = 1$ 
5: end while

```

---

safety levels in the event of a crash. The problem includes five decision variables, and three objectives with zero constraints [14, 16]. The sample size for the same is considered as 54 in this endeavour.

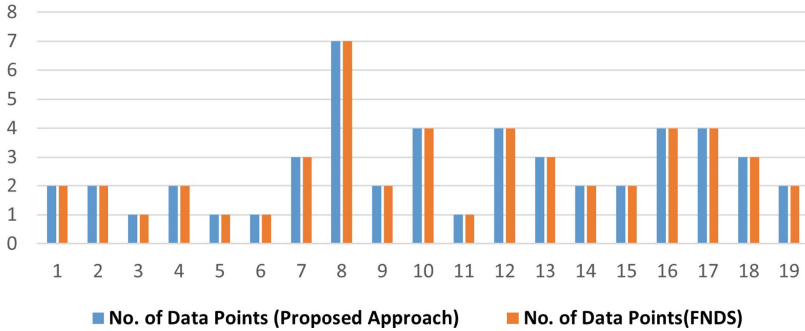
4. **Dataset of Deb-Thiele-Laumanns-Zitzler-1 (DTLZ-1):** DTLZ-1 engages decision variables of  $n$  numbers and can have varying objective functions whose details can be found here [6]. For verifying the proposed mechanism two different samples are considered, the first one has a sample size of 50 with 2 objective functions of maximization type. The second sample consists of a sample size of 500 with 5 objective functions of maximization type.

### 3 Result and Discussion

Following the implementation of the proposed method across various benchmark datasets, the outcomes achieved are at par with those obtained through the FNDS of NSGA-III and NSGA-II. This suggests that the proposed approach can effectively substitute the FNDS component in NSGA-III and NSGA-II or in that matter wherever non-dominated sorting is required, primarily due to its capability to manage streamed data, a feature lacking in other state-of-the-art algorithms. Figure 2 demonstrates the number of Fronts generated after applying the proposed approach on the “DTLZ-1” dataset. For its first sample of size 50 with the number of objective functions being 2, 19 Fronts, each having a different number of solutions are generated, which is similar to the FNDS mechanism for the same sample configuration. The second sample of the “DTLZ-1” dataset features 500 solutions with 5 objective functions of maximization type. For this sample, the count of generated Fronts is 26 as denoted in Fig. 3, this is at par with the FNDS mechanism for the same sample configuration. Figure 4 displays the results obtained from the “LOTZ” dataset, comprising a sample size of 256 and 2 objective functions. The number of Fronts generated and each element residing in different Fronts match the FNDS mechanism with the proposed method. The same thing goes for the “Vehicle Crashworthiness” dataset with a sample size of 54 and the number of objective functions being 3. Here, the number of Fronts generated for both FNDS and the proposed approach is 16 with every Front holding exactly the same solutions for both mechanisms. Figure 4 represents the same. The proposed method has also been tested on additional benchmark datasets, namely, “ZDT-1” and “OneMinmax”. For the “ZDT-1” dataset, two separate samples of sizes 500 and 1000, each with two objective functions, were utilized. In both cases, only one Front was generated, housing all solutions, which

aligns with the outcomes observed with the FNDS mechanism. Similarly, for the “Oneminmax” dataset, only one Front was generated, encompassing all solutions for a sample size of 128.

**Fronts vs No. of Data Points in each Front for DTLZ - 1**



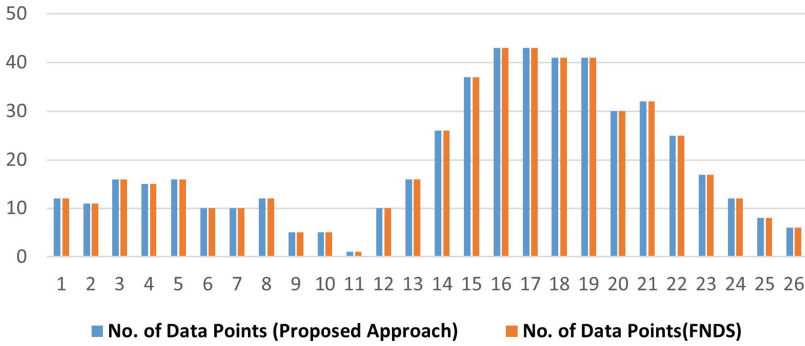
**Fig. 2.** Count of generated Fronts and count of objective space points in each Front, after utilizing the linked list structure approach and the FNDS for the DTLZ-1 dataset of sample size 50 and number of objective functions being 2.

### 3.1 Complexity Analysis

After thoroughly analyzing all the comprehensive scenarios illustrated in Fig. 1, it has been determined that the proposed algorithm exhibits an average and worst-case time complexity of  $O(MN^2)$ , while in the best-case scenario, the time complexity is  $O(MN)$ . In addition, its space complexity is  $O(MN)$ . Regarding complexity, it is clear that our approach demonstrates superior performance compared to the efficient existing algorithms, ENS-SS and ENS-BS, specially in the best-case scenario. While our method matches the best-case scenario of the M-Front approach, it surpasses the method in terms of space complexity.

**Analysis of Time Complexity in the Best-Case Scenario.** The best-case scenario for the proposed nested list mechanism arises in two specific situations. Firstly, it occurs when all incoming elements exclusively exhibit *Case – 2* behaviour, as illustrated in Fig. 1. This indicates that all incoming elements only display indifference to each other, resulting in the generation of only one Front for every other point in the objective space. In this scenario, the time complexity is  $O(MN)$ . Secondly, the best-case scenario also arises when the  $(N + 1)^{th}$  element arrives, and there are either  $N$  existing Fronts or nearly  $N$  existing

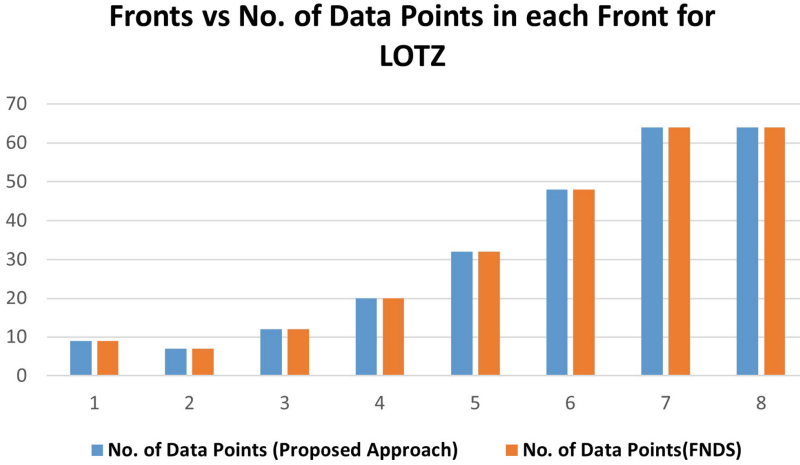
**Fronts vs No. of Data Points in each Front for DTLZ - 1 with 5 objective functions**



**Fig. 3.** Count of generated Fronts and count of objective space points in each Front, after utilizing the linked list structure approach and the FNDS for the DTLZ-1 dataset of sample size 500 and number of objective functions being 5.

Fronts, and the incoming element is dominated by the first or, in the case of exactly  $N$  existing Fronts, by the only existing elements in the Fronts. This scenario of checking dominance relation with the only Front element requires  $O(M)$  time. The overall time complexity in this case remains  $O(MN)$ . Therefore, the proposed mechanism demonstrates a time complexity of  $O(MN)$  in the best-case scenario. However, there are other trivial cases, when  $N = 2$ , only one  $M$  comparison is required which is nothing but the complexity of  $O(M)$ , and when  $N = 1$ , the complexity becomes  $O(1)$ , as depicted in *Case – 1* in Fig. 1.

**Analysis of Time Complexity in the Worst-Case And Average-Case Scenario.** The worst-case and the average-case time complexities coincide in the proposed methodology. In the context of *Case – 1* depicted in Fig. 1, where the incoming element itself forms a new Front and resides there, it is evident that the time complexity for this scenario is  $O(1)$ . Similarly, in *Case – 2* of Fig. 1, where the incoming element exhibits indifference with all the existing elements in a specific Front comprising, say,  $N$  elements, the maximum time complexity is  $O(MN)$ . Moving on to *Case – 3* of Fig. 1, if we consider the worst-case scenario, where the incoming element dominates all the elements in a particular Front, there could be  $N$  elements already present in that Front, all of which are dominated by the incoming element. In this scenario, all the dominated elements are placed in a queue, requiring  $O(N)$  time complexity, as  $N$  elements are being queued. Additionally, since the incoming element checks for dominance with all  $N$  elements in the Front, it necessitates  $O(MN)$  time complexity. Moreover, when the displaced  $N$  elements, kept in the queue, form a



**Fig. 4.** Count of generated Fronts and count of objective space points in each Front, after utilizing the linked list structure approach and the FNDS for the LOTZ dataset of sample size 256 and number of objective functions being 2.

new Front immediately after their previous one, it results in a time complexity of

$$M(1 + 2 + 3 + 4 + \dots + N) = \frac{MN(N + 1)}{2} \equiv O(MN^2).$$

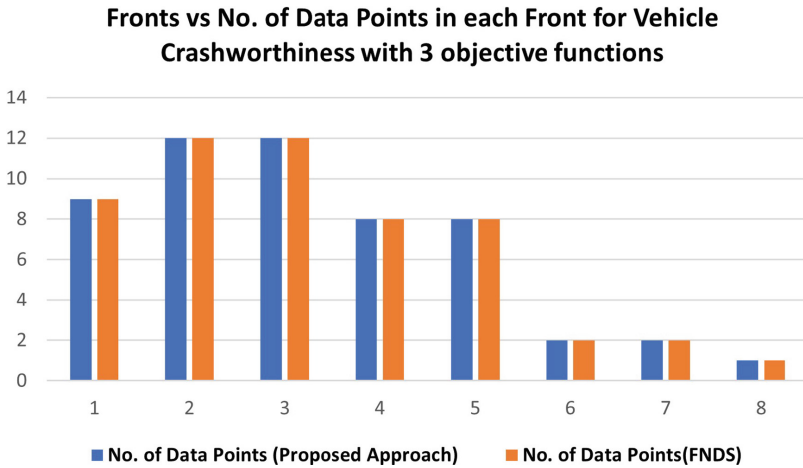
The same logic applies to *Case – 4* demonstrated in Fig. 1, where the incoming element can dominate all but the last element in a Front of  $N$  elements, resulting in a time complexity of  $O(MN^2)$ . Finally, for *Case – 5* in Fig. 1, the required time complexity is  $O(MN)$ . So to sum up, the time complexity of the proposed mechanism in the worst-case scenario is

$$(O(1) + O(MN) + O(MN^2) + O(MN)) \equiv O(MN^2).$$

**Space Complexity Analysis.** The mechanism employs a nested linked list structure and a queue to store displaced dominated objective space points, which are then assigned to their respective Front(s). Thus, if  $N$  points are contained within a single Front,  $O(MN)$  is the space complexity. Similarly, suppose there is more than one Front, each containing a different number of elements even when some element is displaced to the queue. In that case, the total space complexity remains  $O(MN)$ , as the sum of the number of elements across all Fronts, including the queue, is nothing but  $N$ .

### 3.2 Correctness and Completeness

In addition to effectively handling the streamed data points in the objective space, the suggested mechanism can generate Fronts at par with the FNDS,



**Fig. 5.** Count of generated Fronts and count of objective space points in each Front, after utilizing the linked list structure approach and the FNDS for the Vehicle Crashworthiness problem of sample size 54 and number of objective functions being 3.

as previously discussed, suggesting its correctness. Moreover, the mechanism is not restricted to predefined or limited numbers of objective functions; it can accurately operate with any number of objective functions. Another noteworthy aspect of the proposed approach is its ability to handle identical data points without issues. Each incoming element is placed into a distinct Front in a non-overlapping manner, highlighting the robustness of the mechanism.

## 4 Conclusion

The content outlined in this article introduces a novel, accurate, comprehensive, and simple approach for handling streamed data points within a multi-objective framework. This methodology, with minimal representation in current literature, proves to be both straightforward and impactful for performing the non-dominated sorting of streamed data points with space complexity being  $O(MN)$ , since it is not possible to work without holding all the  $N$  number of data points. In the best-case scenario, it shows great promise with a complexity of  $O(MN)$ , this is at par with the best best-case complexity in literature, since Li et al. consider trivial cases for its computation, while for the worst-case scenario, it shows a complexity of  $O(MN^2)$  which is not worse than the existing state-of-the-art mechanisms. In the average case scenario,  $O(MN^2)$  is the complexity. While our research group is actively investigating various methods to address the challenge of non-dominated sorting for streamed data points, the approach presented here, employing a linear structure, is one such promising mechanism. Despite its promise, computationally, it is incapable of outperforming the existing algorithms in the worst-case scenario, though the existing algorithms work



only for static data points. We aim to delve deeper and explore additional avenues to enhance the computational efficiency in the worst case of the proposed mechanism.

**Acknowledgement.** The authors express their gratitude to the Department of Computer and System Sciences, Visva-Bharati, Santiniketan, for facilitating a conducive research environment. One of the authors also acknowledges the University Grants Commission, Ministry of Education, Government of India, for the financial support received through the UGC NET JRF (SRF) scheme, with reference no.: 200510331778.

## References

1. Bao, C., Xu, L., Goodman, E.D., Cao, L.: A novel non-dominated sorting algorithm for evolutionary multi-objective optimization. *J. Comput. Sci.* **23**, 31–43 (2017)
2. Burlacu, B.: Rank-based non-dominated sorting. arXiv preprint [arXiv:2203.13654](https://arxiv.org/abs/2203.13654) (2022)
3. Coello, C.A.C.: *Evolutionary algorithms for solving multi-objective problems*. Springer (2007). <https://doi.org/10.1007/978-0-387-36797-2>
4. Deb, K.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL report 200001 (2000)
5. Deb, K.: *Multi-objective optimization using evolutionary algorithms*, vol. 16. John Wiley & Sons (2001)
6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 1, pp. 825–830. IEEE (2002)
7. Deb, K., Tiwari, S.: Omni-optimizer: a procedure for single and multi-objective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) *Evolutionary Multi-Criterion Optimization*, pp. 47–61. Springer, Berlin, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-31880-4\\_4](https://doi.org/10.1007/978-3-540-31880-4_4)
8. Drozdik, M., Akimoto, Y., Aguirre, H., Tanaka, K.: Computational cost reduction of nondominated sorting using the M-front. *IEEE Trans. Evol. Comput.* **19**(5), 659–678 (2014)
9. Fang, H., Wang, Q., Tu, Y.C., Horstemeyer, M.F.: An efficient non-dominated sorting method for evolutionary algorithms. *Evol. Comput.* **16**(3), 355–384 (2008)
10. Fortin, F.A., Grenier, S., Parizeau, M.: Generalizing the improved run-time complexity algorithm for non-dominated sorting. In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pp. 615–622 (2013)
11. Gustavsson, P., Syberfeldt, A.: A new algorithm using the non-dominated tree to improve non-dominated sorting. *Evol. Comput.* **26**(1), 89–116 (2018)
12. Jensen, M.T.: Reducing the run-time complexity of multi objective EAs: The NSGA-II and other algorithms. *IEEE Trans. Evol. Comput.* **7**(5), 503–515 (2003)
13. Li, K., Deb, K., Zhang, Q., Zhang, Q.: Efficient nondomination level update method for steady-state evolutionary multi objective optimization. *IEEE Trans. Cybern.* **47**(9), 2838–2849 (2016)
14. Liao, X., Li, Q., Yang, X., Zhang, W., Li, W.: Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Struct. Multidiscip. Optim.* **35**, 561–569 (2008)

15. Mandal, S., Dutta, P.: Multi-objective non-overlapping front generation: a pivot-based deterministic non-dominated sorting approach. In: Maji, P., Huang, T., Pal, N.R., Chaudhury, S., De, R.K. (eds.) *Pattern Recognition and Machine Intelligence: 10th International Conference, PReMI 2023, Kolkata, India, December 12–15, 2023, Proceedings*, pp. 559–567. Springer Nature Switzerland, Cham (2023). [https://doi.org/10.1007/978-3-031-45170-6\\_58](https://doi.org/10.1007/978-3-031-45170-6_58)
16. Mandal, S., Dutta, P., Ghosh, A.: Decomposition into multi-objective fronts: a pivot-based deterministic non-dominated sorting approach. In: Bhattacharyya, S., Banerjee, J.S., Köppen, M. (eds.) *Human-Centric Smart Computing: Proceedings of ICHCSC 2023*, pp. 55–66. Springer Nature Singapore, Singapore (2024). [https://doi.org/10.1007/978-981-99-7711-6\\_5](https://doi.org/10.1007/978-981-99-7711-6_5)
17. McClymont, K., Keedwell, E.: Deductive sort and climbing sort: new methods for non-dominated sorting. *Evol. Comput.* **20**(1), 1–26 (2012)
18. Mishra, S., Saha, S., Mondal, S., Coello, C.A.C.: A divide-and-conquer based efficient non-dominated sorting approach. *Swarm Evol. Comput.* **44**, 748–773 (2019)
19. Moreno, J., Rodriguez, D., Nebro, A.J., Lozano, J.A.: Merge nondominated sorting algorithm for many-objective optimization. *IEEE Trans. Cybern.* **51**(12), 6154–6164 (2020)
20. Roy, P.C., Islam, M.M., Deb, K.: Best order sort: a new algorithm to non-dominated sorting for evolutionary multi-objective optimization. In: *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 1113–1120 (2016)
21. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **2**(3), 221–248 (1994)
22. Tang, S., Cai, Z., Zheng, J.: A fast method of constructing the non-dominated set: arena’s principle. In: *2008 Fourth International Conference on Natural Computation*, vol. 1, pp. 391–395. IEEE (2008)
23. Wang, H., Yao, X.: Corner sort for pareto-based many-objective optimization. *IEEE Trans. Cybern.* **44**(1), 92–102 (2013)
24. Xue, L., Zeng, P., Yu, H.: SETNDS: a set-based non-dominated sorting algorithm for multi-objective optimization problems. *Appl. Sci.* **10**(19), 6858 (2020)
25. Zhang, X., Tian, Y., Cheng, R., Jin, Y.: An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **19**(2), 201–213 (2014)
26. Zhang, X., Tian, Y., Cheng, R., Jin, Y.: A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans. Evol. Comput.* **22**(1), 97–112 (2016)
27. Zhang, X., Tian, Y., Jin, Y.: Approximate non-dominated sorting for evolutionary many-objective optimization. *Inf. Sci.* **369**, 14–33 (2016)
28. Zheng, W., Liu, Y., Doerr, B.: A first mathematical runtime analysis of the non-dominated sorting genetic algorithm II (NSGA-II). In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 10408–10416 (2022)
29. Zhou, Y., Chen, Z., Zhang, J.: Ranking vectors by means of the dominance degree matrix. *IEEE Trans. Evol. Comput.* **21**(1), 34–51 (2016)
30. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.* **8**(2), 173–195 (2000)



# Mapping the Unknown: A New Approach to Open-World Video Recognition

César D. Parga<sup>1</sup>✉, Xosé M. Pardo<sup>1</sup>, and Carlos V. Regueiro<sup>2</sup>

<sup>1</sup> Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS),  
Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain  
{cesardiaz.parga,xose.pardo}@usc.es

<sup>2</sup> CITIC, Computer Architecture Group, Universidade da Coruña,  
15071 A Coruña, Spain  
carlos.vazquez.regueiro@udc.es

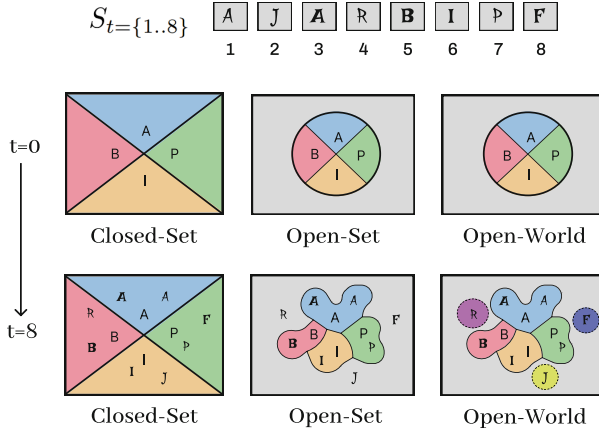
**Abstract.** Intelligent agents must strive to (re)map the constantly changing environment in which they operate, in order to remain adaptive and efficient. In open-world recognition (OWR) a system has to: detect new emerging categories, recognize new instances of known categories, and continually update knowledge based on the data streams it receives, mostly unannotated. In this work, we propose a hybrid method to deal with OWR that combines deep feature embeddings with dynamic ensemble methods for a continuous reshaping of boundaries in feature space. Our approach is flexible to update to patterns in the border of what is already known (concept-drift), detect and create models for new categories, recover from mistakes, and mitigate catastrophic forgetting, even in semi-supervised contexts. As an application use case, we have considered the problem of semi-supervised video face recognition, where the spatial-temporal coherence is harnessed to augment data. Our experiments show that the system responds adequately to the unknowns, adding models for new identities, and improving its performance.

**Keywords:** Ensemble learning · Incremental Learning · Open-World · Instance Recognition

## 1 Introduction

With the general technological advancement, more and more robust models are being designed to operate out of the lab in real-world conditions. In these settings, it is quite often unrealistic to assume the availability of large, unbiased, domain-specific datasets of annotated samples before the start of training.

In general real contexts, data are received in streams, at variable pace, and their distribution, as the environment conditions, are non-stationary [26]. Moreover, if data are unlabeled, the challenge is to distinguish between data drifts and samples belonging to new categories. Usually, collecting a labeled dataset from stream data is expensive. This more general setting is known as open-world recognition (OWR) [7].



**Fig. 1.** In open world recognition, an agent must detect new categories (letters F, R and J in toy example) while being able to maintain, or even improve, the recognition capability on the already known ones (letters A, I, B and P).

Data streams usually contain samples belonging to already known classes/instances but also samples of unknown ones. A query detected as unknown should be treated as a known from that point forward. This OWR scenario contrasts with the ones of closed-set and open-set [7] (Fig. 1).

Video surveillance offers a genuine context for OWR, because of the continuous appearance of new individuals (known and unknown) at different times, with streams of data captured in non-stationary conditions. Besides, by taking advantage of spatial-temporal continuity in video sequences, it is possible to mitigate the paucity of annotated data by performing a kind of data augmentation.

In incremental learning models, the main concern is the loss of knowledge during adaptation to changes in data distributions. They tend to overfit to the new incoming data, rather than gaining in generalization. We propose a new approach based on a set of dynamic ensembles, where base classifiers are joined as ensemble members to enhance its recognition accuracy. Dynamically modeling the frontiers of short clusters of representative samples allows identifying unknown identities and enrolling them (initialization of a new ensemble), without assumptions on data distributions.

Our method allows the adaptation to new knowledge, dealing with problems such as catastrophic forgetting or erroneous updates. In this work we deal with semi-supervised learning where a system is feed with few labeled data, and its predictions on queries are used as pseudo-labels in the incremental learning process. The contributions of this paper can be summarized in:

- A semi-supervised incremental learning approach designed for instance learning in open-world operation with video sequences.
- A method based on dynamic ensembles and the theory of extreme values to distinguish between concept-drift occurrence and unknown detection.

- A strategy to continuously remap the feature space by refining the frontiers of the already known and including new boundaries upon the detection of unknowns.

The rest of the paper is organized as follows. First, we survey the related work and describe our approach. Finally, we present results and main conclusions.

## 2 Related Work

**Incremental Learning (IL).** IL is the ability of a model to gradually learn from new examples without resorting to a complete retraining, while preserving previous knowledge [38,40]. IL tackles the problem of changing environments [36], and includes instance-IL [3], attribute-IL [13], class-IL [25,27], and task-IL [9]. In the realm of artificial neural networks, different strategies have been proposed to mitigate the (catastrophic) forgetting of previous knowledge [10,33]: 1) regularization schemes [2]; 2) parameter isolation methods [18,32], and 3) replay or memory based mechanisms [1,17].

**Stream Learning.** Within continual learning, there are two major paradigms: incremental batch learning, and stream learning [33]. Static approaches make use of large datasets divided into batches for offline training. Usually they can be easily adapted to operate in the incremental batch learning regime [22,34], where mini batches are accumulated during system operation for future retraining cycles. Stream learning [16] takes incremental batch learning a step further: the batch size is equal to one, and phases of training and operation are intertwined to make use of new data as quickly as possible. This paradigm can be used in supervised, semi-supervised and unsupervised modes.

In *semi-supervised learning*, only a small amount of labeled data are available. SAND [15] uses a semi-supervised ensemble classifier and detects concept drift on classifier confidence estimates. SENCForest [31] uses a unique ensemble of multiclass isolation trees as a detector of unknown classes.

In *unsupervised learning*, all data are unlabeled, so common approaches are based on clustering. References about deep networks that deal with unlabeled data are scarce [24]. The most common mechanisms to deal with stream classifiers are Decision Trees [29] and ensemble methods [35].

**Open World Recognition (OWR).** A related problem to OWR is the incremental learning of new emerging classes, which allow including new classes dynamically, but it is based on the implicit or explicit knowledge of their data labels [8,39]. In OWR, streams of samples of different unknown categories can be interleaved, so they are usually clustered before modeling of new categories can take place, and the risk of pooling failures must be managed. Nearest Neighbor based classifiers can be adapted to the OWR context as they are not based on training, but they assume specific data distributions. Clustering methods often assume that the cluster centers remain stable and that the clusters are balanced in size, however some alternatives have been proposed that are amenable to the OWR context. In [21], a contrastive clustering for samples of unknown categories is based on their structural sparsity. A hierarchical agglomerative method

(Finch) for the non-parametric clustering within the unknown category pool was proposed in [20]. In [4], a dynamic Gaussian mixture model (GMM) was proposed to continuously update the mean and covariance of each category (the method is easily adaptable to OWR). CBCL [5] uses a pretrained CNN to generate feature embeddings, and a centroid-based concept learning. Since class centroids are independent of each other, the decrease in overall classification accuracy is not catastrophic when new classes are learned. In CBCL-PR [6] each class is represented by a GMM, and a pseudo-rehearsal mechanism is introduced to drawn pseudo-exemplars, which mixed with the real feature vectors of the new classes, are used to retrain a linear classifier in each increment. Alternatively, EVM models are also adaptable to the OWR context (iEVM [23], FMEVM [20]).

In contrast to aforementioned methods, our model uses a dynamic set of ensembles to facilitate the acquisition of new categories and the improvement of existing ones. Besides, any prior assumption was not made about identities or data probability distribution. Finally, it learns on-the-job and does not have a differentiated learning and testing phase.

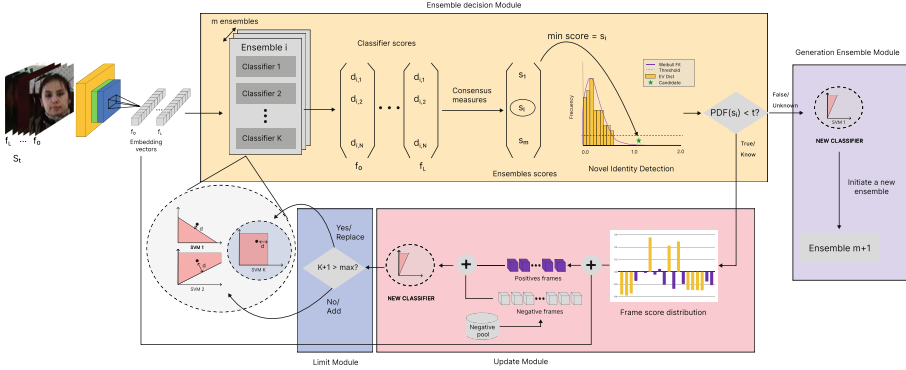
### 3 Dynamic Ensembles for OWR

Our proposal is based on a combination of known and novel techniques. It consists of a dynamic set of ensembles (one per identity) where base classifiers are joined/replaced to upgrade models of specific identities without interfering with each other. Ensembles isolate updates, and make changes reversible, thus circumventing the catastrophic forgetting problem. Our method is initiated with video sub-sequences of a few identities as the only semi-supervision. It makes use of the theory of extreme values (EVT) to distinguish between a concept-drift and the emergence of a new category.

Our approach to OWR in video surveillance (Fig. 2) allows the adaptation to changes in appearance of identities due to endogenous or exogenous factors. The system consists of a hybrid architecture, with an embedding module  $\varepsilon(\cdot)$ , based on deep neural networks, and a classification module based on ensembles which base classifiers are exemplars of SVM (e-SVM).

Our proposal takes advantage of transfer learning from large labeled datasets to get feature embeddings that feed the instance-incremental learning layers. Features embeddings are provided by the last pair of convolutional and batch normalization layers of the frozen ResNet100-ArcFace (RN100-AF) network trained on MS1MV2 dataset [11]. This encoding transforms face crops into 512-D vector.

Learning is semi-supervised, in the sense that true labels are only provided to create the initial base classifier for each of a small number of ensembles (known identities). Initially known, and new identities are modeled by dynamic ensembles (Fig. 2). Each classifier is trained using specific exemplar views of an identity. An ensemble is updated whenever a concept-drift is detected (Ensemble Decision Module). When query embeddings are close to boundaries defined by the ensemble with the highest score, a decision about concept-drift/unknown is done. If drift is detected, a few frame embeddings of the subsequence are selected to train



**Fig. 2.** Sequence feature embeddings are presented to the Ensemble Decision Module. The most probable identity is assigned based on ensemble scores. If confidence is below a threshold, it must be decided whether a drift has occurred or the identity is unknown. In case of drift, a new e-SVM is built (Update Module) and added to the corresponding ensemble. An old classifier could be replaced depending on the availability of computer resources (Limit Module). If the identity is unknown, a new ensemble is initiated (Generation Ensemble Module).

a new base classifier. When the identity is detected as unknown, a new ensemble is initiated with one base classifier [28].

The key of the learning is to update to patterns in the border of what is already known, and initiate a new ensemble when a query has been detected as unknown. Ensembles endow the system with flexibility to recover from mistakes and to mitigate catastrophic forgetting. However, to account for memory usage concerns, the number of base classifiers must be limited, which leads to forgetting some of the previous knowledge during replacements.

### 3.1 Ensemble Decision Module

Ensemble decisions are based on the normalized scores provided by base classifiers, e-SVMs in our implementation (Ensemble Decision Module in Fig. 2). Both feature embeddings and classifier outputs are normalized in our method. Each feature embedding  $x_i$  is divided by its L2 norm to give a normalized representation  $s_i$ , while the output of each linear e-SVM classifiers  $y^j(s_i)$  is also normalized to give values in range  $[-1, 1]$ .

The decision at ensemble level,  $y_e(s_i)$ , is computed as the median of the scores provided by e-SVMs, which is equivalent to a majority voting decision:

$$y_e(s_i) = \text{median}(y_e^0(s_i), y_e^1(s_i), \dots, y_e^{z-1}(s_i)) \tag{1}$$

The sequence score at ensemble  $E_e$  level,  $Y_e$ , is computed based on the median of the frames' scores. We take advantage of the temporal coherence assumption to assign a unique identity to the whole input sequence.

$$Y_e = E_e(S) = \text{median}(y_e(s_0), y_e(s-1), \dots, y_e(s_{L-1})) \tag{2}$$

**Algorithm 1.** Ensemble Decision.

---

**Input:**  $S = \{s_0, s_1, \dots, s_{L-1}\}$  query sequence embeddings  
 $Y = E(S) = \{Y_0, Y_1, \dots, Y_{L-1}\}$  query sequence scores  
**Parameters:**  $T_w$  Weibull threshold  
 $E = \{E_0, E_1, \dots, E_{M-1}\}$  current set of ensembles  
**Output:**  $ID$  inferred identity

- 1:  $y_m = \min(Y)$
- 2: **if**  $y_m < T_s$  **then**
- 3:      $ID = \arg(y_m)$
- 4: **else**
- 5:      $m = \text{median}(Y \setminus \{y_m\})$
- 6:      $V = \{(m - x) \mid x \in (Y \setminus \{y_m\}) \wedge (x < m)\}$
- 7:     Fit  $V$  to a Weibull function  $W$
- 8:     **if**  $W(m - y_m) < T_w$  **then**
- 9:          $ID = \arg(y_m)$
- 10:    **else**
- 11:        $ID = M$  (unknown)
- 12:    **end if**
- 13: **end if**
- 14: **return**  $ID$

---

The use of median function is more robust than made decisions under a subset of maximum or minimum values, due to the function is not sensible to outliers as the maximum or minimum could be in a context of larger sequences.

The ensemble scores,  $Y$ , is the vector with the responses provided by all the current  $M$  ensembles for the same sequence  $S$ :

$$Y = \{Y_e\}_{e=0}^{M-1} \quad (3)$$

The decision at supra-ensemble level determines the identity of the video-sequence based on  $Y$ . The strategy followed in this case is two-folded. First, to be recognized as a known identity, the minimum  $y_m$  of all scores  $Y$  for the query sequence needs to be below a threshold  $T_s$ . Otherwise, a check has to be done to distinguish between a concept-drift or an *unknown* identity.

**Concept-Drift or Unknown.** To tackle this problem, a decision mechanism was implemented based on EVT, which has been widely used for reliability applications, as well as outlier detection.

As any input sequence belongs to a unique identity, the ensembles associated with other identities should deliver non-match outputs. According to the Fisher-Tippet-Gnedenko Theorem of EVT, for the case of left-bounded positive samples, the distribution of the extreme values is given by the Weibull distribution [28]:

$$W(x; \mu, \alpha, \beta) = \frac{\alpha}{\beta} \left( \frac{x - \mu}{\beta} \right)^{\alpha-1} e^{-\left(\frac{x-\mu}{\beta}\right)^\alpha}, \quad (4)$$



---

**Algorithm 2.** Generation of a New Ensemble.

---

**Input:**  $S = \{s_0, s_1, \dots, s_{L-1}\}$  current query embeddings**Parameters:**  $E = \{E_0, E_1, \dots, E_{M-1}\}$  set of ensembles $N = \{n_0, n_1, \dots, n_{Q-1}\}$  pool of negative sample embeddings $positive \subset S$  set of positive embeddings

- 1:  $w^0 \leftarrow$  e-SVM training using *positive* as positive samples and 100 random negative samples from  $N$
  - 2:  $E_m = \{w^0\}$  new generated ensemble
  - 3:  $E = E \cup E_m$
- 

where  $\mu \in \mathfrak{R}$ ,  $\alpha \in \mathfrak{R}$  and  $\beta \in \mathfrak{R}$  are locations, shape, and scale parameters. To avoid the problem of dealing with negative values of scores, we perform the variable change  $\hat{x} = m - x$ , being  $m$  the median of all scores except  $y_m$ , and discarding negative values (the furthest from the tail of the distribution).

Analyzing all the ensembles' scores, an unknown identity can be detected following Algorithm 1. To apply this method we need to initiate the system with at least 10 known identities, since at least 5 values are needed to fit the parameters of a Weibull distribution, and we consider half of them, the closest to  $y_m$ . A threshold  $T_W$  is used to distinguish between concept-drift of the known ( $argc(y_m)$ ) and the unknown. This way, score  $y_m$  can be checked whether it comes from the Weibull extreme value distribution or not. It is important to note that  $T_w$ , together with  $T_s$ , are the only hyperparameters used by our model.

### 3.2 Generation Ensemble Module

Once an unknown is detected, the Generation Ensemble Module (Fig. 2) creates a new ensemble for this new identity. The new ensemble is initiated with a single base classifier (e-SVM), following the procedure described in Algorithm 2.

### 3.3 Update and Limit Module

Once a sequence is recognized as belonging to a known identity, the Update Module (Fig. 2) builds a new e-SVM if concept-drift is detected. Among the frames in the subsequence, those on the boundaries of the already known are selected to train the new classifier. These are slightly different from what is already known, which enables the possibility of unsupervised learning, and keeps low the risk of including samples of different identities in the training of the new classifier.

On its part, the Limit Module is responsible for keeping the size of each ensemble within limits. A new classifier is always added when a concept drift is detected, but when limits are reached, an old classifier is selected to be replaced. Replacement processes are based on the assessment of the diversity of base classifiers. The classifier with the lowest contribution to ensemble diversity measure is

**Algorithm 3.** Update and Limit Module.

---

**Input:**  $E_{ID}$  ensemble of the recognized ID  
 $S = \{s_0, s_1, \dots, s_{L-1}\}$  sequence embeddings  
 $Y = \{y_{ID}(s_0), y_{ID}(s_1), \dots, y_{ID}(s_{L-1})\}$  scores at frame level  
**Parameters:**  $W = \{w^0, w^1, \dots, w_{Z-1}\}$   
 $lim$  = maximum ensemble size  
 $N = \{n_0, n_2, \dots, n_{Q-1}\}$  pool of negative samples  
 $P = \emptyset$  set of positive samples to train classifier

- 1:  $I = \text{argmin}(\text{abs}(Y))$  indexes of frames scores closest to 0 (decision boundaries)
- 2:  $P = S[I[0]..I[5]]$
- 3:  $w_a \leftarrow$  e-SVM base classifier trained using positive samples set  $P$  and 100 negative samples drawn from  $N$
- 4: **if**  $Z \leq lim$  **then**
- 5:    $E_{ID} = E_{ID} \cup \{w_a\}$
- 6: **else**
- 7:    $w_r = \text{argmin}D(w^k), \forall k \in \{0..Z-1\}$
- 8:    $E_{ID} = (E_{ID} \setminus \{w_r\}) \cup \{w_a\}$
- 9: **end if**

---

removed. Given an ensemble of classifiers  $E = \{w^i\}_{i=0}^{z-1}$  and subsequence embeddings  $\{s_0, s_1, \dots, s_{L-1}\}$ , diversity is computed as:

$$D(w^i) = \sum_{k=0, k \neq i}^{z-1} d(w^i, w^k) \quad (5)$$

$$d(w^i, w^k) = -\frac{1}{L} \sum_{l=0}^{L-1} \text{sgn}(w^i(x_l)) \cdot \text{sgn}(w^k(x_l)) \quad (6)$$

where  $w^i(x_l)$  is the score of the e-SVM classifier  $w^i$  for the frame embedding  $x_l$  and  $\text{sgn}(\cdot)$  is the sign function.  $d(w^i, w^k)$  gives the correlation between pairs of classifiers, and  $D(w^i)$  is the measure of global correlation of a classifier. The higher the value of  $D$  the higher the diversity provided by the classifier. The full process is described in Algorithm 3.

## 4 Experiments Preliminary

In order to assess the accuracy of our approach to open world recognition in video surveillance, we have used three of the few appropriated and publicly available annotated video datasets. Aspects as the quality of the frames and the completeness of the video sequences available have a crucial impact on the performance of the assessed approaches [14].

The datasets are: Face COX [19], Face-in-Action (FiA) [12], and YouTube-Faces [37]. COX, designed for assessment of video surveillance approaches, includes 3 cameras with different poses, environmental illumination and resolution conditions, and subsequences of 1000 identities of interest (IoIs). FiA

simulates the capture of frames at passport control, with better resolution than the COX dataset. It includes subsequences of 6 cameras at 3 points in time of different IoIs, varying their number from 214 (first time) to 153 (last time). YouTubeFaces, widely used in video face recognition, contains 3425 videos of 1595 IoIs. Each identity appears in several sequences that go from 1 to 6.

#### 4.1 Experiment Dataset Configuration

In experimentation, each subsequence is a short incoming video stream. Subsequences used for evaluation consist of frames captured from different cameras in each dataset. Firstly, 10 subsequences for each individual were generated. Secondly, these subsequences were divided into three groups: 1) **labeled**, used to create initial ensembles (first 5 frames), 2) **unlabeled**, used during system **operation** (8 subsequences with 20 frames per identity), and 3) **test**, one with 20 frames per identity to evaluate the system’s performance. These requirements, give a lower number of identities available for each dataset: 746 in COX, 162 in FiA, and 344 in YouTubeFaces.

Hereafter, each sequence is denoted by  $S_t^k$ , where  $t$  refers to temporal order and  $k$  to the identity. Accordingly,  $t = 0$  corresponds to the subsequences to create the initial ensembles,  $t = 1..8$  correspond to operation subsequences and subsequences  $t = 9$  will be used to test the system.

#### 4.2 Experimental Setup

First, models of the initial IoI are created, which consist of ensembles with only one base classifier (e-SVM). These base classifiers are created with the initial training subsequences,  $S_0^k$ , as positive samples and 100 embeddings of a set of negatives from other initial IoIs (randomly drawn from a pool of samples). A similar procedure is followed for the classifiers added to ensembles during system operation. The remaining 8 subsequences,  $S_{\{1..9\}}^k$ , will vary in length from 5 to 20 frames, depending on the available video sequences of each individual.

The experimental process was organized in adaptation steps, after which a measurement of system performance was done. An adaptation step corresponds to a complete iteration over the  $k$  available identities, at the same point in time  $t$ . In addition to this, it will iterate 3 times over that integer set of steps  $t = \{1..8\}$ . Repeating the data several times allows us to increase the number of update steps as well as to see how the system behaves with redundant data. The maximum number of classifiers per ensemble is set at 10.

Algorithm 4 describes the process that was repeated 10 for each experiment. Identities were presented in a different and random order in each of these runs. The metrics used are Precision, Recall, and F1.

This analysis will make it possible to observe whether the behavior of the system remains stable after the addition of new ensembles and the impact it has on known individuals. In order to make the data comparable with experiments carried out in an open set context, it is necessary to set a limit to those new identities that have been added.

---

**Algorithm 4.** Experimental procedure.
 

---

**Input:**  $S_t^k$  sequence  $t$  of the identity  $k$   
**Parameters:**  $f$  #sub-sequences per identity (excluding the initial)  
 $L$  #iterations  $N_i = \text{\#initial IoI}$   
 $N_a = \text{\#IoI to add}$   $N_U = \text{\#identities in the universe}$   
 $R = \text{\#remaining identities to add}$

- 1: **for**  $k = 0$  **to**  $N_i - 1$  **do**
- 2:   Initialize  $E_k$  with  $W_k^0$  based on  $S_0^k$
- 3: **end for**
- 4: Perform testing using the set of  $\{S_f^k\}_{k=0}^{N_i}$
- 5:  $R = N_a$
- 6: **for**  $lap = 0$  **to**  $L - 1$  **do**
- 7:   **for**  $t = 1$  **to**  $f - 1$  **do**
- 8:     **for**  $k = 0$  **to**  $N_U - 1$  **do**
- 9:       Perform adaptation using  $S_t^k$
- 10:       **if**  $S_t^k \notin \text{known IoI}$  and  $R > 0$  **then**
- 11:          Initialize ensemble  $k$  using  $S_0^k$
- 12:           $R = R - 1$
- 13:       **end if**
- 14:     **end for**
- 15:     Testing the full IoI set  $\{S_f^k\}_{k=0}^{N_i+N_a}$
- 16:   **end for**
- 17: **end for**

---

## 5 Experiment Results

In this section, we study the impact of progressively adding new identities to the initial IoI set (open-world scenario). Experiments were carried out for different sizes of  $N_i$  and  $N_a$  (Algorithm 4), as shown in Table 1.

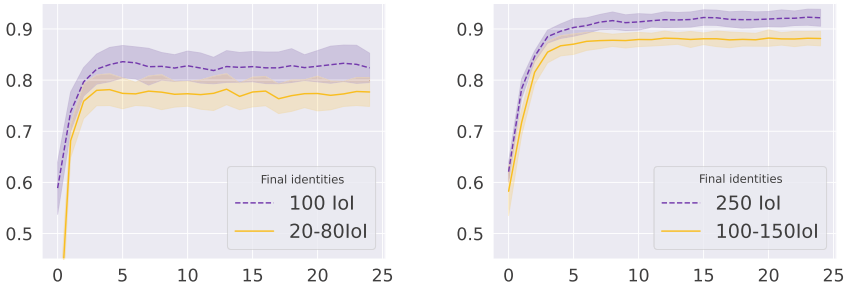
The experiments with COX were carried out in OW for two cases: 20+80 and 100+150. In FiA database there are only 100 identities, so we included the case 20+30. As baselines, we use the case where all identities (100 or 250) were known, and no new identities were introduced, although the decision about known-unknown was made, as well.

Figure 3 illustrates how the recognition capacity of the system evolves in two scenarios on COX dataset when all sequences are repeated 3 times. In both cases, the performance raises quickly in the early stages (first repetition) and then stabilizes (second and third repetition). The initial performance in the open-world is lower, but very soon reaches values close to the open-set. Remarkably, performance does not decline over time, demonstrating the system’s robustness to catastrophic forgetting. It is also worth mentioning that the system must be initialized with a few IoI because of the use of dynamic Weibull thresholding. In our experiments, we initiate the system with at least 20 ensembles, each one with a single e-SVM.

In view of the data collected in Table 1, our system running in open world achieves similar results to those that would be obtained running in open set.

**Table 1.** Results on COX and FiA datasets for different numbers of identities (IoI), knowns (K) and unknowns (UK). Cases of UK=0 correspond to open set (OS) contexts, the rest are open world (OW). Performances ( $\mu(\sigma)$ ) are computed from 10 repetitions of experiments varying the IoIs and their order of presentation.

IoI	Dataset	Precision		Recall		F1-score		
		K-UK	Initial	Final	Initial	Final	Initial	Final
COX (OS)	100-0		78.62 (4.27)	78.62 (3.16)	46.10 (5.89)	86.58 (3.15)	58.88 (5.12)	82.38 (2.86)
COX (OW)	20-80		15.48 (3.52)	73.57 (3.37)	59.76 (12.16)	82.33 (2.68)	24.56 (5.36)	77.68 (2.83)
COX (OS)	250-0		95.49 (1.71)	92.52 (1.83)	46.01 (1.87)	91.87 (2.07)	62.09 (1.94)	92.19 (1.65)
COX (OW)	100-150		81.87 (4.05)	89.42 (1.45)	45.50 (5.53)	86.90 (1.91)	58.26 (4.74)	88.13 (1.43)
FiA (OS)	100-0		97.61 (1.21)	92.84 (2.14)	61.49 (6.67)	89.20 (2.29)	75.23 (5.14)	90.95 (1.31)
FiA (OW)	20-80		50.45 (6.47)	90.31 (2.82)	58.89 (9.98)	87.78 (2.32)	54.19 (7.47)	89.01 (2.35)
FiA (OS)	50-0		93.02 (3.06)	83.80 (3.39)	65.95 (5.29)	88.30 (2.74)	77.00 (3.26)	85.95 (2.43)
FiA (OW)	20-30		53.31 (10.6)	80.01 (4.76)	60.00 (13.8)	83.80 (3.85)	56.24 (11.7)	81.81 (3.80)



**Fig. 3.** F1 scores versus learning stages for 100 and 250 IoI in COX dataset: (left) 100 known and 20 known with 80 initial unknown (right) 250 known and 100 known with 150 initial unknown.

Looking at Table 1, we can see that the biggest difference between the values of open set and open world, in terms of F1, is around 5% for the case of COX dataset. The same behavior can be observed for the case of FiA, being the case 20-80 in open world very similar to that of the 100-0 in open set (1.94%).

Analyzing the values of the *completeness* and *homogeneity* metrics, typically used in clustering, we obtain mean values of  $\pm 99\%$  and  $\pm 97\%$ , respectively, in the composition of ensembles at the end of the experiments. These could be considered evidence of the robustness of our approach to generate diverse sets with a low proportion of labeling errors after inference.

## 5.1 Comparison Against State-of-the-Art Face Recognition in OWR

In this section, we compare the performance of our method against a number of proposed solutions for open-world recognition, using the implementations provided in [20]. The initial samples of the 20 known identities have been used to train the classifiers and neural networks necessary for the operation of these models. Processing was performed frame by frame, with data received as a stream.

**Table 2.** Results ( $\mu \pm \sigma$ ) of different methods adapted to OWR context on three datasets, from 10 repetitions of experiments. Best results are bolded.

Datasets →	COX		FiA		YouTubeFaces	
Methods ↓	F1-measure	Accuracy	F1-measure	Accuracy	F1-measure	Accuracy
NNO [7]	4.10 ± 0.20	37.06 ± 4.10	25.02 ± 1.94	33.95 ± 2.03	13.44 ± 4.64	16.29 ± 13.8
NCM [30]	12.06 ± 3.06	89.43. ± 0.19	18.59 ± 1.77	46.17 ± 0.69	15.24 ± 2.94	17.29 ± 1.53
CBCL [5]	19.53 ± 6.97	85.82 ± 1.52	21.70 ± 0.97	46.72 ± 0.50	15.66 ± 0.47	17.29 ± 0.28
GMM [4]	12.10 ± 3.93	89.43 ± 0.25	20.14 ± 1.36	46.87 ± 0.57	11.54 ± 3.42	12.79 ± 0.71
FEVM [20]	26.87 ± 1.79	85.56 ± 0.16	56.85 ± 2.06	54.92 ± 2.31	67.57 ± 1.83	52.71 ± 2.30
Ours	<b>84.35 ± 3.26</b>	<b>96.57 ± 0.67</b>	<b>90.27 ± 1.23</b>	<b>88.99 ± 1.43</b>	<b>90.56 ± 2.19</b>	<b>84.11 ± 3.59</b>

Testing was performed by incorporating 80 known identities to the initial set of 20, setting the number of positive samples at 10 and the number of negatives at 100. In this case, all methods were evaluated on the 3 datasets and with two different performance metrics, accuracy and F1-score. Table 2 shows the results for each of the different models, with the best result for each metric in bold. In all cases, it can be seen how our method offers results significantly better than the ones provided by the second best.

The methods have been adapted to work with video sequences and to make fair comparisons, the same feature embeddings have been used in all methods. Using face-specific embeddings is more convenient than using pretrained models on general object datasets. It should be noted that no modifications have been made to other parts of the original codes.

## 5.2 Sensitivity About Parameters

The impact of the selected positive samples on the creation of new classifiers was evaluated for the case of 20 knowns vs 80/230 unknowns in COX, considering four options: frames with scores closest to zero (“On Boundaries”), first frames in the sub-sequence, frames randomly selected, and those with the highest positive scores. The first provided the best F1-scores (2% better than random).

We also studied the dependence of results on the size  $N$  of negatives dataset for the case of 10 positives. With values in range [25, 500], the highest score was achieved with 500 negatives. However, the difference between the mean F1-scores of 100 and 500 was always less than 1, so we used  $N = 100$  in our experiments.

To assess the impact of  $T_w$ , we measure initial and final performance for the case of 50 IoIs in a universe of 100, for values of  $T_w$  in range [0.001, 0.1]. Results show that overall incremental learning capabilities of our approach are maintained within this range, with a maximum around 0.01.

We perform a study to set  $T_s$  threshold and avoid computation of the Weibull distributions. The greater the number of knowns, the lower the value of  $T_s$  is necessary to maintain good performance. A value of 0.15 is enough conservative;

a value of 0.05 already has a low impact on performance, since the method is quite robust to erroneous inferences. We set a value  $T_s = 0.01$  in all experiments.

Our model achieves significant improvement over well-known outlier detection techniques such as Median Absolute Deviation (MAD) or cosine distance, with an increase  $\pm 40\%$  in F1-score for 500 identities. Furthermore, using twice the number of classifiers per ensemble results in only a 1% improvement in F1-score, due to the good optimization made by the replacement policy.

## 6 Conclusions

We present a novel approach to instance recognition in open-world video applications, emphasizing adaptability to evolving scenarios. By employing dynamic ensembles tailored to each category, our method not only handles an increasing number of unknowns but also improves identification of known identities.

Our system demonstrates continuous improvement in recognition capability, achieving significant F1-scores with limited initial knowledge of identities. Additionally, it remains robust against diverse data and low-resolution streams, making it suitable for real-time video processing. To prevent catastrophic forgetting, our approach dynamically adjusts ensemble composition, ensuring retention of past knowledge while accommodating new information, as Fig. 3 illustrates.

The replacement policy is derived from common diversity metrics. We observe that in only 5% of the cases, ensembles were built using samples of different identities (mistakes in operation). We have never found the case of two ensembles initiated from belonging to the same identity. However, it could be of interest to create a mechanism for merging ensembles that resonate with the same identity, thus making the system more resilient.

Looking forward, we aim to explore spatio-temporal coherence in wider application domains, such as general instance object recognition and anomaly detection. Additionally, we aim to devise methods for detecting base classifiers in different ensembles trained with subsequences of the same real identity, and how to change the value of  $T_s$  according to the evolution of the data distribution.

**Acknowledgements.** This work has received financial support from the Spanish government (project PID2020-119367RB-I00); from the Xunta de Galicia, Consellaría de Cultura, Educación e Ordenación Universitaria (2019–2022 ED431G-2019/04 and ED431G 2019/01, and competitive groups 2021–2024 ED431C 2021/48 and ED431C 2021/30), and from the European Regional Development Fund (ERDF/FEDER). César D. Parga has received financial support from the Xunta de Galicia and the European Union (European Social Fund - ESF).

## References

1. Acharya, M., Hayes, T.L., Kanan, C.: RODEO: replay for online object detection. In: The British Machine Vision Conference (BMVC) (2020)

2. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: learning what (not) to forget. In: European Conference on Computer Vision (ECCV), pp. 139–154 (2018)
3. Anowar, F., Sadaoui, S.: Incremental learning framework for real-world fraud detection environment. *Comput. Intell.* **37**(1), 635–656 (2021)
4. Arandjelovic, O.D., Cipolla, R.: Incremental learning of temporally-coherent gaussian mixture models. In: British Machine Vision Conference (BMVC) (2005)
5. Ayub, A., Wagner, A.R.: Cognitively-inspired model for incremental learning using a few examples. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 222–223 (2020)
6. Ayub, A., Wagner, A.R.: CBCL-PR: a cognitively inspired model for class-incremental learning in robotics. *IEEE Trans. Cogn. Dev. Syst.* **15**(4), 2004–2013 (2023)
7. Bendale, A., Boulton, T.: Towards open world recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1893–1902 (2015)
8. Cai, X.Q., Zhao, P., Ting, K.M., Mu, X., Jiang, Y.: Nearest neighbor ensembles: an effective method for difficult problems in streaming classification with emerging new classes. In: IEEE International Conference on Data Mining (ICDM) (2019)
9. Davidson, G., Mozer, M.C.: Sequential mastery of multiple visual tasks: networks naturally learn to learn and forget to forget. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9279–9290 (2020)
10. De Lange, M., et al.: A continual learning survey: defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(7), 3366–3385 (2022)
11. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4690–4699 (2019)
12. Goh, R., Liu, L., Liu, X., Chen, T.: The CMU Face In Action (FIA) Database. In: Zhao, W., Gong, S., Tang, X. (eds.) *Analysis and Modelling of Faces and Gestures: Second International Workshop, AMFG 2005, Beijing, China, October 16, 2005. Proceedings*, pp. 255–263. Springer, Berlin, Heidelberg (2005). [https://doi.org/10.1007/11564386\\_20](https://doi.org/10.1007/11564386_20)
13. Gorraeb, S., Rejab, F.B.: IK-prototypes: incremental mixed attribute learning based on k-prototypes algorithm, a new method. In: *Intelligent Systems Design and Applications*, pp. 880–890 (2021)
14. Guo, G., Zhang, N.: A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* **189**, 102805 (2019)
15. Haque, A., Khan, L., Baron, M.: SAND: semi-supervised adaptive novel class detection and classification over data stream. In: *AAAI Conference on Artificial Intelligence* (2016)
16. Hayes, T.L., Cahill, N.D., Kanan, C.: Memory efficient experience replay for streaming learning. In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9769–9776 (2019)
17. He, J., Mao, R., Shao, Z., Zhu, F.: Incremental learning in online scenario. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13923–13932 (2020)
18. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Lifelong learning via progressive distillation and retrospection. In: *European Conference on Computer Vision (ECCV)*, pp. 452–467 (2018)
19. Huang, Z., et al.: A benchmark and comparative study of video-based face recognition on cox face database. *IEEE Trans. Image Process.* **24**(12), 5967–5981 (2015)



20. Jafarzadeh, M., Dhamija, A.R., Cruz, S., Li, C., Ahmad, T., Boulton, T.E.: Open-world learning without labels. CoRR abs/2011.12906 (2021)
21. Joseph, K.J., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5830–5840 (2021)
22. Kemker, R., Kanan, C.: FearNet: brain-inspired model for incremental learning. In: International Conference on Learning Representations (ICLR) (2018)
23. Koch, T., Liebezeit, F., Riess, C., Christlein, V., Kohler, T.: Exploring the open world using incremental extreme value machines. In: International Conference on Pattern Recognition (ICPR), pp. 2792–2799 (2022)
24. Ksieniewicz, P., Woźniak, M., Cyganek, B., Kasprzak, A., Walkowiak, K.: Data stream classification using active learned neural networks. *Neurocomputing* **353**, 74–82 (2019)
25. Kukleva, A., Kuehne, H., Schiele, B.: Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9020–9029 (2021)
26. Kulkarni, R., Revathy, S., Patil, S.: An empirical study of online learning in non-stationary data streams using ensemble of ensembles. *Int. J. Adv. Sci. Eng. Inf. Technol.* **11**, 1801 (2021)
27. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: multi-class incremental learning without forgetting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
28. Lopez-Lopez, E., Pardo, X.M., Regueiro, C.V.: Incremental learning from low-labelled stream data in open-set video face recognition. *Pattern Recogn.* **131**, 108885 (2022)
29. Manapragada, C., Webb, G.I., Salehi, M.: Extremely fast decision tree. In: International Conference on Knowledge Discovery and Data Mining, pp. 1953–1962 (2018)
30. Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Metric learning for large scale image classification: generalizing to new classes at near-zero cost. In: European Conference on Computer Vision (ECCV), pp. 488–501 (2012)
31. Mu, X., Ting, K.M., Zhou, Z.H.: Classification under streaming emerging new classes: a solution using completely-random trees. *IEEE Trans. Knowl. Data Eng.* **29**(8), 1605–1618 (2017)
32. Ostapenko, O., Puscas, M., Klein, T., Jähnichen, P., Nabi, M.: Learning to remember: a synaptic plasticity driven framework for continual learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11313–11321 (2019)
33. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: a review. *Neural Netw.* **113**, 54–71 (2019)
34. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: iCaRL: incremental classifier and representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
35. Wankhade, K.K., Jondhale, K.C., Dongre, S.S.: A clustering and ensemble based classifier for data stream classification. *Appl. Soft Comput.* **102**, 107076 (2021)
36. Wei, X.S., Ye, H.J., Mu, X., Wu, J., Shen, C., Zhou, Z.H.: Multi-instance learning with emerging novel class. *IEEE Trans. Knowl. Data Eng.* **33**(5), 2109–2120 (2021)
37. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 529–534 (2011)

38. Wu, Z., Baek, C., You, C., Ma, Y.: Incremental learning via rate reduction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1125–1133 (2021)
39. Zhou, D.W., Yang, Y., Zhan, D.C.: Learning to classify with incremental new class. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(6), 2429–2443 (2022)
40. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5867–5876 (2021)



# ESL: Explain to Improve Streaming Learning for Transformers

Meghna P. Ayyar<sup>(✉)</sup>, Jenny Benois-Pineau, and Akka Zemmari

LaBRI, CNRS, Univ. Bordeaux, UMR 5800, 33400 Talence, France  
{meghna-parameswaran.ayyar, jenny.benois-pineau,  
akka.zemmari}@u-bordeaux.fr

**Abstract.** AI systems in real-world scenarios must be able to learn continuously from a stream of data while adapting quickly to concept drift. We propose a training strategy called Explain to improve Streaming Learning (ESL) for the online streaming learning setting where the models have to learn from data on the fly in a single pass. ESL leverages model explanations to identify salient input regions, guiding the streaming learner to focus on these regions by masking the non-salient ones during training. In this work, we focus on using transformers for streaming learning and adapt our CNN-based Feature Explanation Method (FEM) [13] to propose Rollout-FEM with ESL for transformers. We validate the ESL strategy for our streaming learners Entropy-based Move-To-Data (EMTD), its variant EMTD with re-targetting (EMTDR), and the state-of-the-art streaming learning method ExStream [14] and benchmark it on two streaming learning datasets and a real-world egocentric video dataset. Our experiments demonstrate that training with the explanation-based ESL strategy has a better performance than standard training, and EMTDR with ESL achieves the best performance compared to ExStream across the datasets.

**Keywords:** Streaming Learning · Online Continual Learning · Explainable AI

## 1 Introduction

Deep Neural Networks (DNNs) have shown exceptional success for multiple tasks when trained on large-scale, well-annotated datasets with fixed data distribution. However, real-world learning scenarios are dynamic with evolving data. Humans can accumulate and improve their learning to adapt to these changes. Ideally, machine learning algorithms, particularly bio-inspired models, should mirror a similar adaptability. Traditional machine learning schemes become ineffective in non-stationary environments where drifts in the data distribution cause concept drifts, resulting in misclassification by the model. Incremental learning thus emerged as a crucial paradigm to allow models to learn from new data while retaining previous knowledge and has gained popularity in recent years [27].

Class incremental learning, especially, has gained traction, where the models gradually learn new categories of data [21].

The authors of [16,22] argue that the class incremental learning scenario is limited by the random and discrete shifts in the distribution of the benchmarking data. Each learning step introduces a new set of non-repeating classes, which fails to represent the gradual nature of change observed in realistic environments, where previously seen classes can reappear with new contextual variations. Consequently, the Online Streaming Learning (OSL) scenario becomes important wherein the model has to learn from individual data samples available on the fly in a single pass [14]. In particular, we focus on the common scenario where the streaming data undergoes a concept drift [8] e.g. in the case of egocentric videos recorded from a wearable camera [20]. In such data streams, the data distribution might gradually shift over time for already observed categories due to changes in the background, lighting, data collection method like the exact position of a wearable camera, etc. Therefore, it is essential that the model adapts to the new environment quickly rather than its ability to recognize “old” data.

Moreover, as AI and Deep Learning models are becoming ubiquitous, the need for transparency and improving their trustworthiness has become mandatory. Multiple methods have been proposed to explain the decisions of Deep Neural Networks (DNNs). In recent years, a new focus of research has emerged for leveraging eXplainable AI (XAI) techniques to intervene in the behaviour of machine learning models. This is achieved by introducing additional supervision signals or prior knowledge obtained from explanations into the model reasoning process [31]. In this paper, we propose the Explain to improve Streaming Learning (ESL) strategy to use an XAI method to improve OSL.

We recently proposed the Entropy-based Move-To-Data (EMTD) [6] method for OSL. It updates only the weights of the final classifier layer to adapt the model to the evolving relation between the input and the output, i.e., under concept drift. EMTD is faster than the state-of-the-art ExStream [14], as it updates the weights without gradient computation. As the update is not in the optimal direction of the gradient, a model drift is observed after a few EMTD updates. EMTDR (EMTD with Retargeting) [6] reduces the model drift by using a small subset of recent samples from the data stream for a single gradient update to adjust the parameters of the model. In this work, we couple EMTDR with ESL and validate it on two benchmark streaming datasets and a real-life challenging dataset of egocentric datasets. We also compare it with EMTD and the SOTA ExStream to demonstrate the effectiveness of ESL for streaming learning. The main contributions of our work are as follows:

- We adapt the FEM explanation method [13] for ESL with Vision Transformer (ViT) [10] based models to introduce the Rollout-FEM (RFEM) method to get the importance maps for the data samples.
- Propose the ESL strategy to identify and enhance the important regions in the input using the RFEM for ViT-based models to improve streaming learning.

Though ESL can be used with other methods, we focus on transformers in this work due to their superior performance when compared to models like Convolutional Neural Networks (CNNs) and its “self-attention” mechanism that can capture long-range dependencies amongst the features of the input.

The rest of the paper is organized as follows. Section 2 presents an overview of the related work for streaming learning and the use of explanation methods to improve training. Section 3 presents our ESL strategy with details about the RFEM method and a brief overview of our continual learning methods EMTD and EMTDR. Section 4 gives the details of the experiments, results, and the ablation study. Section 5 concludes the work and outlines future research perspectives.

## 2 Related Work

In **Online Streaming Learning** (OSL), data arrives sequentially, one sample at a time. In Incremental Batch learning, data arrives in batches, and the model can access only the current batch of data but can iterate over each batch multiple times during learning. In contrast, OSL operates on a single-instance basis, where the model learns in a single pass. Streaming learning methods like ExStream [14] and REMIND [15] use a buffer to store a subset of past samples, which are replayed alongside the new samples to train the classifier layers of the model. We currently focus on streaming learning with a fixed taxonomy for incoming samples but with a concept drift in the data. For images and videos, concept drift occurs with gradual appearance changes, variations in lighting and backgrounds, shifts in camera viewpoints and different views for the same visual content. It leads to a specific type of concept drift called covariate drift, where the relation between the input and output remains the same but the distribution of the input data evolves with time [8].

Test-Time Adaptation (TTA), specifically Continual Test-Time Domain Adaptation (CTDA) methods like [30], are similar to OSL as both adapt a pre-trained network to new target domains without access to the source domain. However, CTDA adapts the model during inference time without ground truth labels, whereas OSL adapts the model during training with real labels. CTDA methods usually rely on pseudo-labels, which can lead to error accumulation, and the methods often use different strategies to improve the quality of these pseudo-labels. For example, CoTTA [29] uses augmentation-averaged predictions to reduce error accumulation and stochastic restoration of some neurons at each step to reduce forgetting. Another method, TENT [28], only adapts the parameters of the BatchNorm layers of the model using entropy minimization of the predictions. In contrast, OSL methods have access to new samples with labels and thus avoid errors due to pseudo-labels, ensuring more accurate adaptation.

In our previous work [6], we introduced a family of fast OSL methods based on the ‘Move-To-Data’ (MTD) principle. The MTD method adjusts the weights of the final classification layer for the new input from the data stream. Our Entropy-based MTD (EMTD) method selectively updates the model using only

the samples with high information from the stream, outperforming the SOTA method ExStream [14]. As EMTD directly updates the weights of the neuron in the final layer without gradient-based optimisation, it leads to a model drift after a few updates. To counter this, we introduced a conditional retargeting (EMTDR) with a single update in the direction of the gradient using a small subset of recent samples used for EMTD updates.

**Explanation Methods:** Many methods have been proposed for the explanation of CNNs including popular white-box gradient-based methods like Grad-CAM [24] and Vanilla Backpropagation [25]. These methods are class-specific and leverage gradients with respect to the model’s predictions for a class to explain the contribution of input features to the network’s decision. The authors of [3] offer a comprehensive overview and categorization of the many recent methods proposed for the explanation of DNNs. Shapley Additive Explanation (SHAP) proposed by [17], is a gradient-free method that uses cooperative game theory principles to compute Shapley values. These values quantify the contribution of each feature to a model’s prediction by using a linear model to assess how the prediction changes when certain features are included or excluded from the input. Feature Explanation Method (FEM) [13] is our gradient-free and class-agnostic method that identifies the “strong” features from the maps of the final convolutional layer of a CNN. To identify the “strong” features, FEM assumes that the features in the final convolutional layer have a Gaussian distribution. Thus, it uses K-sigma thresholding to select the “rare” and “important” features from the layer. FEM then uses a linear combination of these maps with the channel importance weights to get the final explanation map.

In comparison, fewer methods focus on the explanations for Transformer models, often using the ‘self-attention’ maps of the layers as explanations [26]. Attention Rollout [1] is a class-agnostic method that recursively multiplies the attention maps of each transformer layer to get the final visualization of the importance of the different input tokens. I-SAW is a class-specific method [18] that weights the attention map of each layer with the attention gradient w.r.t the label class followed by the rollout to improve the visualizations. Furthermore, [2] extend Layer-wise Relevance Propagation (LRP) [7] for transformers using relevance propagation rules to propagate relevance scores from the output to the input. In our study, we propose adapting the FEM method for transformers by integrating it with the Rollout method to create the class-agnostic Rollout-FEM (RFEM) for transformers.

**Using Explanations for Training:** In their recent work [31], the authors describe approaches that use explanations to improve model generalization and reasoning. Some popular methods include i) data augmentation using explanations by changing the distribution, such as using spectral analysis-based explanation to remove biased or ‘poisoned’ samples from the input [4], ii) intermediate feature augmentation in the network to reinforce relevant parameters while masking nonrelevant ones, and iii) augmenting the loss function with additional explanation supervision [19]. The latter combines attention and class-specific

attention gradients to provide extra supervision to the loss function for the training of a transformer.

Further, some works have used explanations for training in an incremental setting. [12] uses an LRP-based neural freezing for incremental learning. It reduced the plasticity of neurons important for previous tasks by assigning them a lower learning rate or completely freezing them. However, its iterative computation of LRP on the test set of current tasks reduces its efficiency and adaptability to complex architectures. In contrast, Remembering for the Right Reasons (RRR) [11] uses a rehearsal-based strategy for continual learning by storing visual explanation maps generated by the Grad-CAM method [24] for each sample in a memory buffer. RRR enforces consistency between the explanation maps generated during training and those in the memory buffer. [23] also uses Grad-CAM [24] maps to crop and store the most salient patch of the input image itself in the memory for class-incremental learning. Nevertheless, using fixed-size cropping like [23] or class-specific guidance like [18] could be detrimental while streaming learning with changes in the data distribution. Moreover, fixed-size of the cropping may lack robustness with objects of different sizes or if the dataset requires some context information to be retained. In this work, we propose a *class-agnostic* strategy to identify and retain the most important region of the input images. Our method uses an XAI method with a dynamic cropping size regulated by the strength of the transformer attention for the given image.

### 3 Proposed Framework: ESL

The ESL strategy comprises three steps: i) Use an XAI method to obtain the importance map for the input, ii) Dynamically select the salient patch from this importance map, and iii) Train a streaming learner using the transformed input. While the ESL strategy applies to most Deep Neural Networks (DNNs), we currently focus on Vision Transformer (ViT) [10] models.

#### 3.1 XAI Method: Rollout Feature Explanation Method (RFEM)

Vision transformers decompose the input images into a sequence of square patches and embedded into lower-dimensional vectors called query  $Q$ , key  $K$ , and value  $V$  for each patch in every transformer layer. The self-attention  $A$  is computed as  $A = Q \cdot K^T$ , allowing the model to determine the importance of each patch in relation to others. It enables the model to focus on relevant regions of the input image during training.

*Attention Rollout* [1] is a class-agnostic attention visualization method that aggregates the attention weights for all the layers  $L$  as shown in Eq. 1 where  $I$  is the identity matrix for residual connections and  $H$  is the total attention heads within a layer  $l$ . Instead of simply averaging the attention of all heads  $H$  in a transformer layer  $l$ , we propose to use our XAI method FEM [13] to identify

only the “strong” and thus important attentions for the aggregation across the heads.

$$A^l = I + \sum_{h=1}^H A_h^l, \quad A_{roll} = \prod_{l=1}^L A^l \quad (1)$$

Thus, we first aggregate across the layers by a recursive multiplication similar to rollout as shown in Eq 2.

$$A_h^l = I + A_h^l, \quad \forall h = 1, \dots, H, \quad A_{h,roll} = \prod_{l=1}^L A_h^l \quad (2)$$

$K$ -sigma thresholding with  $K = 1$ ,  $\mu_h$  as the mean and  $\sigma_h$  is the standard deviation of the values is applied to these maps  $A_{h,roll}$ , as shown in Eq. 3, as in the baseline FEM [13], to get the binary maps  $m_h$  with only the “rare” and thus “strong” attentions:

$$m_h(A_{h,roll}) = \begin{cases} 1 & \text{if } a_{i,h} \geq \mu_h + K * \sigma_h \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

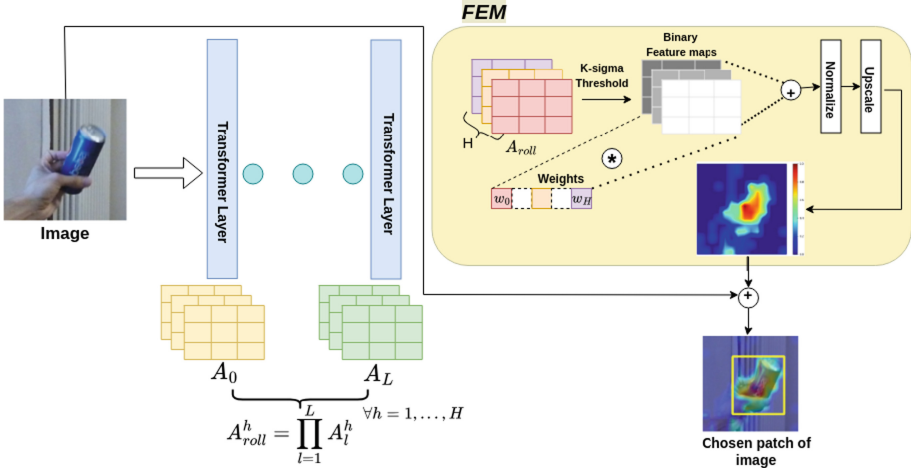
A linear combination with the mean  $\mu_{h,roll}$  of the attention maps  $A_{h,roll}$  as weights is then used to combine the binary maps  $m_h(A_{h,roll})$  into a single map  $M(x)$  for the input  $x$ .  $M(x)$  is then normalized and resized to the input size to obtain the final RFEM importance map  $M_{RFEM}(x)$ .

### 3.2 Input Patch Selection

For the given input  $x$ , its map  $M_{RFEM}(x)$  is used to retain only the regions of higher saliency while masking others. Contrary to [23] who use a fixed-size cropping of the neighbourhood around the most salient pixel, we propose a *dynamic* approach. The size of the crop is determined by the span of high values of saliency in the maps  $M_{RFEM}(x)$ . When the  $K$ -sigma threshold is applied to the RFEM maps, binary maps are generated to retain only the regions of higher saliency. The salient region with the largest area is identified in each map, and its bounding box (BB) is chosen for cropping. Thus, we get a variable-size input for the transformer, but implement it by masking the pixels in images and video frames outside the salient region by 0. Figure 1 illustrates the detailed steps of RFEM and the subsequent dynamic cropping, shown by the yellow square, for the input image.

In addition to images, we also use this framework for videos using the TimeSFormer [9] based Pooling Transformer [20]. This model has a divided spatial and temporal attention computation to speed up training. Thus, the two attentions are first combined as outlined in [32], followed by RFEM to get per-frame saliency maps for the input video (Fig. 1).





**Fig. 1.** Framework for ESL to select the salient region in the image using Rollout-FEM (RFEM)

### 3.3 Streaming Learner: Entropy-Based Move-To-Data (EMTD) and Retargeting (EMTDR)

The next step of ESL is to use the transformed input from the previous stage to train the model during streaming learning. Our streaming learning methods, EMTD [5] and EMTD with Retargeting (EMTDR) [6] update the model by slightly changing the weights of the last classifier layer, as shown in Eq. 4. Here  $w_t^c$  are the weights of the output neuron for the class  $c$ ,  $v_t$  is the feature vector of the new sample  $x_t$  extracted from the model and  $0 < \varepsilon < 1$ . In a DNN, the activation of the output neuron for the  $c$ -th class  $\hat{y}_t^c = \langle w_t^c, v_t \rangle$ , is high when  $w_t^c$  is close to  $v_t$ . Thus, we "move" the weights  $w_t^c$  in the direction of the feature vector  $v_t$  of the new sample to adjust only the last classifier layer of the model. This update is fast as it does not rely gradient calculation to adjust the weights [5].

$$w_{t+1}^c = w_t^c + \varepsilon (\|w_t^c\| * \frac{v_t}{\|v_t\|} - w_t^c), \tag{4}$$

In the streaming setting, data arrives one sample at a time, and not every sample might be important for updating the model. Thus, we use entropy-based sampling to identify the most informative samples from the data stream for model updates (EMTD). EMTD uses two buffers for this purpose: a small fixed-size buffer  $B_v = \{v^1, v^2, \dots, v^b\}$  of size  $b$  to store the features ( $v$ ) and  $B_e = \{e(v^1), e(v^2), \dots, e(v^b)\}$  to store the entropy of each feature in  $B_v$ . Since the feature extractor layers of the model remain unchanged during learning, we only store the features of new samples to reduce memory. Here, entropy signifies the uncertainty of the model in classifying the sample and is computed using Eq. 5, where  $C$  is the total classes in the dataset and  $p^c$  is the softmax output for all

$c \in \{1, \dots, C\}$ , from the classification layer. The entropy is maximal when the class probabilities  $p^c$  are equal, indicating the network’s uncertainty in classifying the sample. Consequently, we select the sample with the highest entropy at each step to update the neuron of the last layer using Eq. 4.

$$e(v^i) = - \sum_{c=1}^C p^c \log(p^c) \quad (5)$$

When the buffer is full, we check if the feature vector of the incoming sample has a higher entropy than the feature vector with the lowest entropy currently stored in the buffer. If the condition is true, we replace it with the new feature vector; otherwise, we replace the feature vector used for the EMTD update in the previous step.

As the updates with EMTD do not follow the optimal direction of the gradient, a model drift [5] is observed after several updates. Methods like [8] detect a concept drift when model performance deteriorates below a threshold. Building on this, we use a small validation set of samples to monitor the model performance and do a conditional “retargeting” step when the performance drops below a threshold, as shown in Eq. 6. Here, the parameter  $\alpha$  monitors the performance drop, where  $Acc^*$  is the best validation accuracy from *steps* 1 to  $t - 1$  and  $Acc_t$  is the validation accuracy on current step  $t$ . When a drift is detected, the model is retargeted with a single gradient-descent update for the last layer with the samples from our retarget buffer  $B_r$ .  $B_r$  has a fixed small size  $r$  and stores the last  $r$  high entropy samples selected for EMTD. If the condition for retargeting is satisfied, these features are used for a single gradient update of the last classifier layer (EMTD with retargeting EMTDR) to realign the model. Subsequently, the retarget buffer is cleared after the retargeting.

$$\frac{(Acc^* - Acc_t)}{Acc^*} > \alpha \quad (6)$$

## 4 Experiments and Results

### 4.1 Experimental Details

**Evaluation Datasets:** We validated ESL on two benchmark datasets (CoRES50 and Stream51) and a challenging real-world video dataset (BIRDS).

1. **CoRE50** [16] is a continual learning dataset consisting 10 classes of objects recorded during 11 sessions with varying backgrounds. We sampled the 15-second videos at a rate of 5 fps and the final training set comprises approximately 23,982 images, and the test set contains 8,995 images.
2. **Stream51** [22] is a large-scale dataset tailored for streaming learning with images for 51 distinct object classes. It includes 150,736 images in the training set and 2,550 images in the test set.

3. **BIRDS**: Bio-Immersive Risk Management System (BIRDS) dataset used in [20], is a real-life, in-the-wild dataset designed for detecting risk situations among frail individuals from egocentric videos. It has 19,500 videos with five risk categories: environmental risk of fall, risk of domestic accident, physiological risk of fall, risk of dehydration, risk of medication intake, along with a “No Risk” category annotated by expert psychologists. Each video, approximately 2–3s long, is sampled into smaller clips of size  $\Delta_v = 8$  with a stride of 4 frames.

In lifelong learning, an initial dataset is usually available for Phase-0 learning to train the model and establish an initial base initialization. After Phase-0 the feature extractor layers of the model are frozen, and only the last classifier layers are updated during Phase-1 streaming learning. The datasets are split 40% for Phase-0 and 60% for Phase-1 while maintaining the same class distributions. Phase-0 and Phase-1 datasets were further split 80% for training and 20% for validation. The BIRDS dataset has a high class imbalance with ‘No-Risk’ as the majority class. Therefore, 5% of the No-Risk videos were randomly selected for Phase-0, while all the samples were retained for Phase-1 to represent a realistic scenario. The test sets of CoRE50 and Stream51, containing samples from both Phase-0 and Phase-1, are then used to report the final performance of the methods. The results for BIRDS are reported on the Phase-1 validation set as a test set was not available.

**Baseline Methods:** The performances of EMTD and EMTDR with ESL are compared with the following approaches:

1. **Naive** (Fine-tuning without any buffer) The classifier layer of the model is fine-tuned on Phase-1 data, using a batch size of 1 to mimic streaming learning. It serves as the lower bound for the performance of the streaming learning algorithms.
2. **ExStream** [14]: is a SOTA streaming learning method closest to our scenario. It uses a buffer to store the features of the previous and the incoming samples. At each step it uses all the samples in the buffer to update the classifier layers with a single gradient-descent update. It stores  $s$  examples per class in the buffer and merges the two closest features from the buffer for a class to make space for an incoming sample. For all our experiments, we choose  $s = 128$  i.e., 128 samples per class as proposed by the authors.
3. **Offline**: The entire model is conventionally trained using the whole dataset (Phase-0 + Phase-1) and serves as the upper bound for the performance of the model on the given dataset.

**Explanation Methods for ESL:** In addition to the class-agnostic RFEM, we implement ESL with the following SOTA explanation methods and compare their performance:

1. **Attention Rollout** [1] is a *class agnostic* method that aggregates the attentions from different layers of the transformer with a recursive multiplication, as shown in Eq. 1.

2. **Grad-CAM** [24] is a popular *class-dependent* CNN explanation method. It has been adapted for transformers and uses the linear combination of attentions from the last transformer layer (before the classifier head) and class-specific gradient as weights<sup>1</sup> to get the explanation map.
3. **I-SAW** [18] is a *class-dependent* explainer for transformers. It first multiplies the attention maps of each layer by its gradient w.r.t the output class, and similar to rollout, combines the maps with the recursive multiplication, as shown in Eq. 7.

$$A^l = I + \sum_{h=1}^H (\nabla A_h^l * A_h^l), \quad A_{isaw} = \prod_{l=1}^L A^l \quad (7)$$

ViT [10] pretrained on ImageNet1K, with an input size of  $224 \times 224$ , has been used for the *image* datasets CoRE50 and Stream51. The Pooling Transformer [20] with a video input size of  $8 \times 224 \times 224$  has been used for the *video* BIRDS dataset. The parameters for EMTD and EMTDR were set to  $\varepsilon = 0.0002$ , retargeting threshold  $\alpha = 0.0001$ , entropy buffer  $b = 32$  and retarget buffer as  $r = 32$  for the CoRE50 and Stream51 image datasets. The final parameters for BIRDS were set to  $\varepsilon = 0.001$ ,  $\alpha = 0.0001$ ,  $b = 32$  and  $r = 32$ . These values were chosen by grid search and are the same as [6].

## 4.2 Results

**Explanation Methods:** We compare the choice of two class agnostic and two class dependent explanation methods to get the saliency maps using ESL for Phase-0 and Phase-1. For Phase-0, we performed pairwise tests to evaluate the performance of RFEM against other explainers on the image CoRE50 and video BIRDS datasets. Across the three cases, the Wilcoxon signed-rank test yielded a p-value of 0.031 with a five-fold cross-validation. This indicates that ESL with RFEM significantly outperforms ESL with the other considered explainers for Phase-0. Comparing standard training to training with ESL, we observed an increase in accuracy of  $\sim 2$  for CoRE50 and BIRDS datasets.

Table 1 compares the results of the four explanation methods for Phase-1 streaming learning for the three datasets. The results indicate that RFEM is consistently better than other explanation methods across the datasets for the streaming learning methods. To further validate this observation, we did a five-fold cross validation for the explainers on CoRE50 image and BIRDS video datasets. Using the Wilcoxon signed-rank test, we obtained a p-value of 0.031 for RFEM vs the other methods, indicating substantial evidence that RFEM is the better choice. In the streaming setting with concept drift, the relation between the input and the output evolves over time, which the class-dependent methods may fail to capture and could lead to outdated explanations. This could be the reason why the class agnostic method RFEM shows better performance for ESL compared to the class-dependent Grad-CAM and I-SAW methods. The

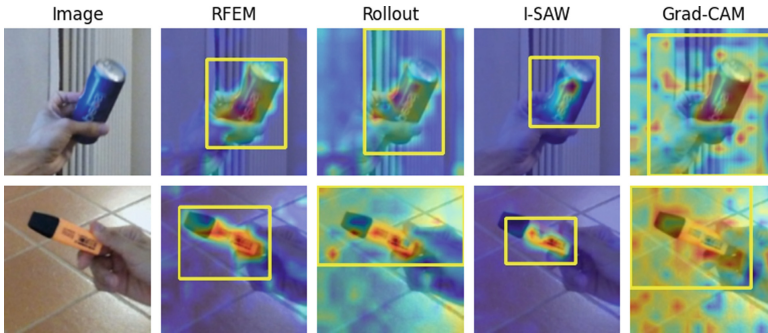
<sup>1</sup> <https://github.com/jacobgil/pytorch-grad-cam/>.

same behaviour is observed for Attention Rollout in most cases, although RFEM consistently outperforms it.

**Table 1.** Phase-1 accuracies of ESL with different explanation methods for the streaming learning methods. RFEM: Rollout FEM (Ours), GC: GradCAM [24], Roll: Attention Rollout [1], IS: I-SAW [18]

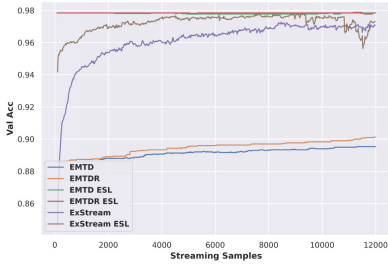
Methods	CoRE50				Stream51				BIRDS			
	RFEM	GC	Roll	IS	RFEM	GC	Roll	IS	RFEM	GC	Roll	IS
EMTD	97.34	94.60	96.40	96.33	94.94	94.07	94.39	94.10	69.14	64.08	68.68	64.68
EMTDR	<b>97.50</b>	95.68	96.69	96.81	<b>95.41</b>	94.42	94.62	94.60	<b>71.20</b>	66.80	70.37	65.37
ExStream	96.79	93.50	94.39	95.20	90.13	89.59	90.02	89.37	56.64	51.19	53.92	53.76

Figure 2 presents a visual comparison of the patches selected by the different explanation methods for images from the CoRE50 dataset. It can be seen that Grad-CAM is not well suited for ESL as the higher saliency regions are more spread out in the map. Attention Rollout maps are slightly better as the regions with high attention are more focused on the objects. The I-SAW maps are more concentrated on the object; however, the weighting of the attention maps with the class-dependent gradients imposes a stronger influence on the attention scores, resulting in partial cropping of the object. In comparison, RFEM maps localize the salient region around the object while also preserving some contextual information.

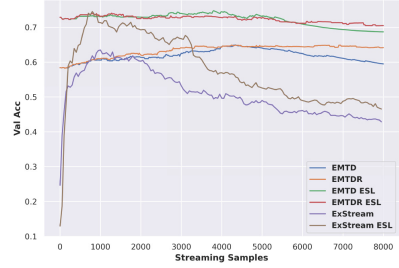


**Fig. 2.** Selection of the patch from CoRE50 dataset for the different explanation methods

**ESL Performance:** Figure 3 illustrates the evolution of accuracies on the Phase-1 validation set during OSL when trained with and without ESL. For the three streaming learning methods-EMTD, EMTDR, and ExStream-training with ESL



(a) CoRe50



(b) BIRDS

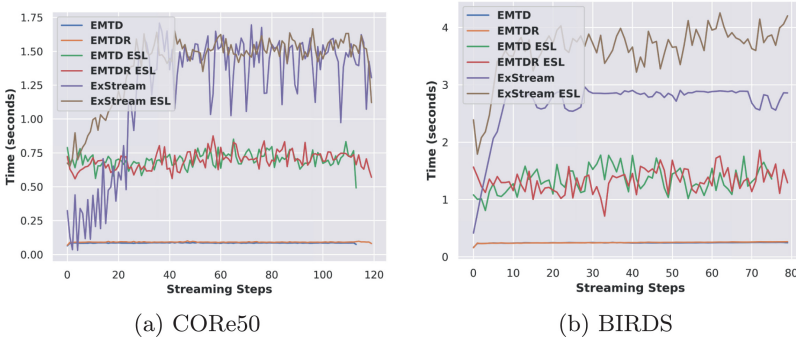
**Fig. 3.** Evolution of validation accuracy for the streaming learners with and without ESL

has a higher performance throughout the learning phase. On CoRE50, the methods show an increasing trend indicating that they learn well on all the new samples. With the BIRDS dataset, we can see that EMTD and ExStream methods have a drop in accuracy after some updates (with and without ESL), which is mitigated by the retargeting in EMTDR. Table 2 compares the final accuracies when training with and without ESL for streaming learning. The full *offline* training is the best achievable performance, while the Naive learning, which updates the model without any corrections, has the lowest results. For the streaming learning methods, EMTD and EMTDR systematically outperform ExStream and EMTDR with ESL has better results in terms of accuracy across the datasets, see line EMTDR in Table 2. Further, training with ESL shows an improvement in performance for all the streaming learning methods. EMTD and EMTDR methods with ESL show an increase of  $\sim 3$  for CoRE50,  $\sim 2$  for Stream51 and  $\sim 8$  for the BIRDS dataset compared to standard training. Notably, both Naive and Offline training show an increase in accuracy across the datasets when trained with ESL indicating that using just the explanation-based input selection works well in both the standard and streaming scenarios. As presented in Sect. 4.1, the test sets for CoRE50 and Stream51 include samples from both Phase-0 and Phase-1. The higher accuracies of our methods, when trained with ESL for these datasets, indicate their improved performance on both old and new data. However, the BIRDS dataset consists of egocentric videos where changes in the camera’s point of view cause the same class to appear differently in the new samples. Thus, it is important for the methods to quickly adapt to new data rather than recognize old data, as it is rarely seen again. Hence, with respect to the stability/plasticity dilemma, we do not study the ability of the updated model to recognize old data. Nevertheless, as training with ESL improves performance on the test sets of CoRE50 and Stream51, we can say experimentally that our methods are stable.

**Time Complexity:** In streaming learning, every new sample on the data stream must be learnt as soon as it is available. Thus, the learner needs to be accurate and fast for the model predictions to align with the changes in the incoming data.

**Table 2.** Phase-1 Accuracies of the streaming learning and the baseline methods for our ESL strategy with RFEM and w/o: without ESL. **Bold:** Upper bound, Blue : best streaming learner

Method	CoRE50		Stream51		BIRDS	
	w/o	ESL	w/o	ESL	w/o	ESL
EMTD	94.05	97.34	94.09	94.94	61.30	69.14
EMTDR	94.2	<b>97.50</b>	94.13	<b>95.41</b>	64.74	<b>71.20</b>
ExStream	92.50	96.79	89.98	90.13	54.46	56.64
Naive	85.60	94.29	84.26	89.88	42.95	55.28
Offline	95.40	<b>98.06</b>	96.23	<b>97.65</b>	67.18	<b>73.76</b>



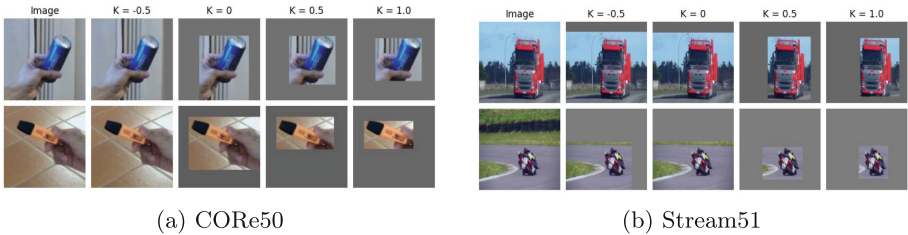
**Fig. 4.** Training time for streaming learners with and without ESL per 100 samples

Figure 4 shows the training times measured for the streaming learning methods EMTD, EMTDR and SOTA ExStream with and without ESL on an image and a video dataset. The times were measured when training using a single NVIDIA A40 GPU. It can be seen that using ESL increases the training time as it does two forward passes through the network, first to calculate the RFEM maps and the second to train on the zero-padded selected patch. But the EMTD and EMTDR methods with ESL are still faster than the basic ExStream. The Wilcoxon signed-pair test for EMTDR + ESL vs. ExStream and Exstream + ESL yielded a p-value lower than 0.01 and thus our method is significantly faster. Although slower than EMTD and EMTDR, training with ESL offers overall improvement in performance while still being faster than SOTA ExStream learning on images. ESL is slower on the BIRDS than CoRE50, as the patches have to be selected for each frame in the input video clip.

### 4.3 Ablation Study

The ablation study for our ESL strategy has been presented by comparing its performance to the standard streaming learning. Additionally, we compared different explanation methods for ESL with transformers. We observed that RFEM

outperforms the other explainers across three datasets for streaming learning. The only parameter tuned for ESL is the threshold  $K$  used to crop the salient patches from the input. As detailed in Sect. 3, the “high” importance scores are determined by statistical filtering with the  $K$ -sigma thresholding rule. Empirically, we determined the appropriate values for  $K$  through a grid search for  $K = [-0.5, 0, 0.5, 1]$ . Figure 5 shows some samples for the CoRe50 and Stream51 datasets for the different values of  $K$ . It can be seen from the figure that lower values of  $K$  result in minimal cropping, while higher values lead to extensive cropping of the input images. Overall, it was observed that  $K = 0.5$  worked well for the image datasets CoRE50 and Stream51, and  $K = -0.5$  was chosen for the BIRDS video dataset to retain more of the background due to the contextual nature of the classes.



**Fig. 5.** The original image and the images after cropping using the RFEM method for different threshold values  $K$  to select the important regions.

## 5 Conclusion and Future Work

Through our ESL strategy, we demonstrated that leveraging explanation methods to identify and enhance salient input regions can improve the performance of transformer models for streaming learning. ESL employs the Rollout-FEM method to generate saliency maps from the self-attention maps of each transformer layer for every new sample encountered in the data stream. We validated our method on two benchmark image datasets and a real-world video dataset to predict risk situations among frail people in a streaming learning scenario with drift in the data distribution. We also compared our RFEM explainer with other explanation methods for transformers. Our results demonstrated that RFEM was better suited for ESL, as it exhibited superior localization of salient regions and outperformed in terms of accuracy.

In addition, we also compared the training times for the streaming learning methods. Though using ESL made EMTD and EMTDR methods slower, they are still faster than the SOTA ExStream method trained without ESL. An interesting future work could consist in using explanation methods that create the class prototypes to monitor how the new data from the stream changes over



time. It could provide further insights on how a streaming learner adjusts to changes in data. It would also be useful to investigate the performance of ESL under Test-time adaptation conditions where the real labels are not available.

## References

1. Abnar, S., Zuidema, W.H.: Quantifying attention flow in transformers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 4190–4197. Association for Computational Linguistics (2020)
2. Ali, A., Schnake, T., Eberle, O., Montavon, G., Müller, K.R., Wolf, L.: XAI for transformers: Better explanations through conservative propagation. In: International Conference on Machine Learning, pp. 435–451. PMLR (2022)
3. Ali, S., et al.: Explainable artificial intelligence (XAI): what we know and what is left to attain trustworthy artificial intelligence. *Inf. Fus.* **99**, 101805 (2023)
4. Anders, C.J., Weber, L., Neumann, D., Samek, W., Müller, K.R., Lapuschkin, S.: Finding and removing clever Hans: using explanation methods to debug and improve deep models. *Inf. Fus.* **77**, 261–295 (2022)
5. Ayyar, M.P., Benois-Pineau, J., Zemmari, A., Amieva, H., Middleton, L.: Entropy-based sampling for streaming learning with move-to-data approach on video. In: 20th International Conference on Content-based Multimedia Indexing, CBMI 2023, Orleans, France, pp. 21–27. ACM (2023)
6. Ayyar, M.P., Poursanidis, M., Benois-Pineau, J., Zemmari, A., Mansencal, B., de Rugy, A.: Family of move-to-data methods for online continual learning for deep neural networks. SSRN 4659402 (2023)
7. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**(7), e0130140 (2015)
8. Bayram, F., Ahmed, B.S., Kassler, A.: From concept drift to model degradation: an overview on performance-aware drift detectors. *Knowl.-Based Syst.* **245**, 108632 (2022)
9. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML, vol. 2, p. 4 (2021)
10. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR (2021)
11. Ebrahimi, S., et al.: Remembering for the right reasons: explanations reduce catastrophic forgetting. *Appl. AI Lett.* **2**(4), e44 (2021)
12. Ede, S., et al.: Explain to Not Forget: defending against catastrophic forgetting with XAI. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) Machine Learning and Knowledge Extraction: 6th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2022, Vienna, Austria, August 23–26, 2022, Proceedings, pp. 1–18. Springer International Publishing, Cham (2022). [https://doi.org/10.1007/978-3-031-14463-9\\_1](https://doi.org/10.1007/978-3-031-14463-9_1)
13. Fuad, K.A.A., Martin, P., Giot, R., Bourqui, R., Benois-Pineau, J., Zemmari, A.: Features understanding in 3D CNNs for actions recognition in video. In: Tenth International Conference on Image Processing Theory, Tools and Applications, IPTA 2020, pp. 1–6. IEEE (2020)
14. Hayes, T.L., Cahill, N.D., Kanan, C.: Memory efficient experience replay for streaming learning. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 9769–9776. IEEE (2019)

15. Hayes, T.L., Kafle, K., Shrestha, R., Acharya, M., Kanan, C.: REMIND your neural network to prevent catastrophic forgetting. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII*, pp. 466–483. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58598-3\\_28](https://doi.org/10.1007/978-3-030-58598-3_28)
16. Lomonaco, V., Maltoni, D.: CORE50: a new dataset and benchmark for continuous object recognition. In: *Conference on Robot Learning*, pp. 17–26. PMLR (2017)
17. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
18. Mallick, R., Benois-Pineau, J., Zemmari, A.: I saw: A self-attention weighted method for explanation of visual transformers. In: *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3271–3275. IEEE (2022)
19. Mallick, R., Benois-Pineau, J., Zemmari, A.: IFI: interpreting for improving: a multimodal transformer with an interpretability technique for recognition of risk events. In: *International Conference on Multimedia Modeling*, pp. 117–131. Springer, Cham (2024). [https://doi.org/10.1007/978-3-031-53302-0\\_9](https://doi.org/10.1007/978-3-031-53302-0_9)
20. Mallick, R., et al.: Pooling transformer for detection of risk events in in-the-wild video ego data. In: *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 2778–2784. IEEE (2022)
21. Pernici, F., Bruni, M., Baccchi, C., Turchini, F., Del Bimbo, A.: Class-incremental learning with pre-allocated fixed classifiers. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6259–6266. IEEE (2021)
22. Roady, R., Hayes, T.L., Vaidya, H., Kanan, C.: Stream-51: streaming classification and novelty detection from videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 228–229 (2020)
23. Saha, G., Roy, K.: Saliency guided experience packing for replay in continual learning. In: *Winter Conference on Applications of Computer Vision, WACV*, pp. 5262–5272. IEEE (2023)
24. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE ICCV*, pp. 618–626 (2017)
25. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualizing image classification models and saliency maps. In: *2nd International Conference on Learning Representations, ICLR, Workshop Track Proceedings* (2014)
26. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
27. Verwimp, E., et al.: CLAD: a realistic continual learning benchmark for autonomous driving. *Neural Netw.* **161**, 659–669 (2023)
28. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726* (2020)
29. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: *Proceedings of IEEE CVPR*, pp. 7201–7211 (2022)
30. Wang, Y., et al.: Continual test-time domain adaptation via dynamic sample selection. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1701–1710 (2024)
31. Weber, L., Lapuschkin, S., Binder, A., Samek, W.: Beyond explaining: opportunities and challenges of XAI-based model improvement. *Inf. Fus.* **92**, 154–176 (2023)
32. Xu, Y.: TimesFormer attention rollout (2022). <https://github.com/yiyixuxu/TimeSformer-rolled-attention>



# Detection of Unknown Errors in Human-Centered Systems

Aranyak Maity<sup>(✉)</sup>, Ayan Banerjee, and Sandeep K. S. Gupta

Arizona State University, Tempe, USA  
{[amaity1](mailto:amaity1@asu.edu), [abanerj3](mailto:abanerj3@asu.edu), [sandeep.gupta](mailto:sandeep.gupta@asu.edu)}@asu.edu

**Abstract.** Artificial Intelligence-enabled systems are increasingly being deployed in real-world safety-critical settings involving human participants. It is vital to ensure the safety of such systems and stop the evolution of the system with error before causing harm to human participants. We propose a model-agnostic approach to detecting unknown errors in such human-centered systems without requiring any knowledge about the error signatures. Our approach employs dynamics-induced hybrid recurrent neural networks (DiH-RNN) for constructing physics-based models from operational data, coupled with conformal inference for assessing errors in the underlying model caused by violations of physical laws, thereby facilitating early detection of unknown errors before unsafe shifts in operational data distribution occur. We evaluate our framework on multiple real-world safety critical systems and show that our technique outperforms the existing state-of-the-art in detecting unknown errors.

**Keywords:** Human-Centered Systems · AI-Safety · Physics Based Surrogate Model

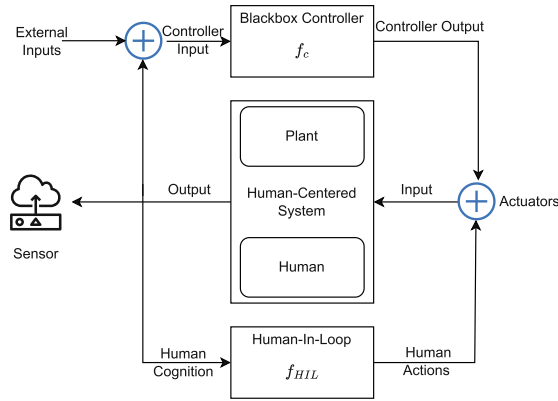
## 1 Introduction

Rapid advancements in Machine Learning (ML) and Artificial Intelligence (AI) have led to an increase in the number of AI-enabled systems being deployed in real-world safety-critical settings. These systems often are deployed in contexts where they can cause potential risks to human participants. It is of utmost necessity to ensure the safety of such Safety Critical Human-Centered Systems and prevent them from causing harm to humans. While substantial efforts have been made to guarantee the safety of these systems, much of the current research focuses on safety assurances during the design phase [7, 8, 10, 18, 19], often overlooking the unpredictable dynamics of real-world settings and the dynamic nature of the human participants. Additionally, while runtime monitoring has

---

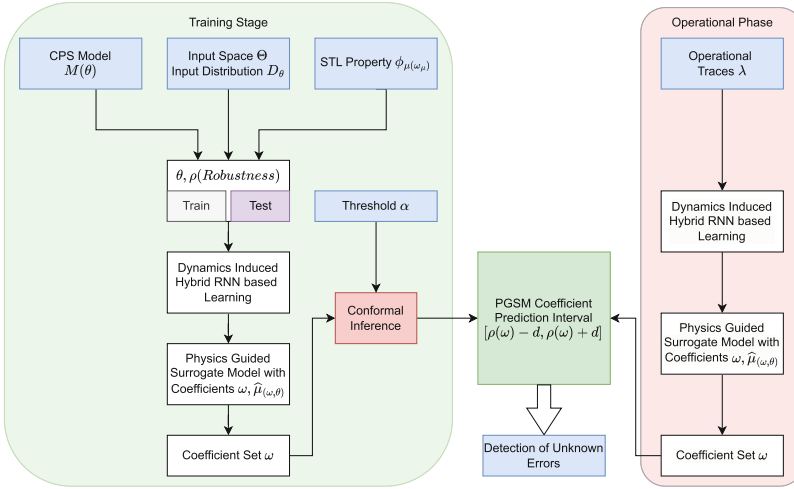
**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_12](https://doi.org/10.1007/978-3-031-78189-6_12).

been explored as a solution to this challenge, existing runtime monitoring techniques [20] need to be trained on the specific errors they are trying to detect, which are often not available. In this paper, **we focus on developing an approach for detecting errors in operational Human-Centered Systems, without prior knowledge of the error signatures.** Our approach of error detection relies solely on the observation of inputs and outputs from the system (Fig. 1). By assuming black-box access (Fig. 1) to the model’s controller, which could be an AI-based controller or a conventional one, such as Model Predictive Control (MPC) we make our detection mechanism model agnostic.



**Fig. 1.** System Model of Human-Centered Systems. In this architecture, the human operator can be both part of the control mechanism and within the operational dynamics of the plant itself. The plant’s state is monitored through sensors and control actions are performed via actuators, processes that are prone to inaccuracies and errors

Recognizing errors in the operational phase presents unique challenges [2–4]. In Human-Centered Systems [5] that are in operation, sensing is limited, and also errors in a component of the systems may not readily have any effect on the trajectories of the sensed variables due to several physical properties. Recently proposed design time stochastic safety verification based on output trajectories [1] may fail to detect errors during operation, since the effect of the errors on the output trajectories (sensor values) may fall within the safe operating conditions. An error may subsequently be combined with known or unknown errors resulting in safety violations with potential fatal consequences [12]. Moreover, in real-world deployments, systems may encounter previously unseen scenarios, many of which are unpredictable and lack predefined error signatures, making it challenging to train machines for their detection. Our approach addresses these challenges by deploying continuous model learning and conformance-checking strategy, focused on the model coefficients that reflect the underlying physical laws governing the system. This strategy is designed to identify structural breaks [17] and deviations indicative of errors, thereby enhancing error detection

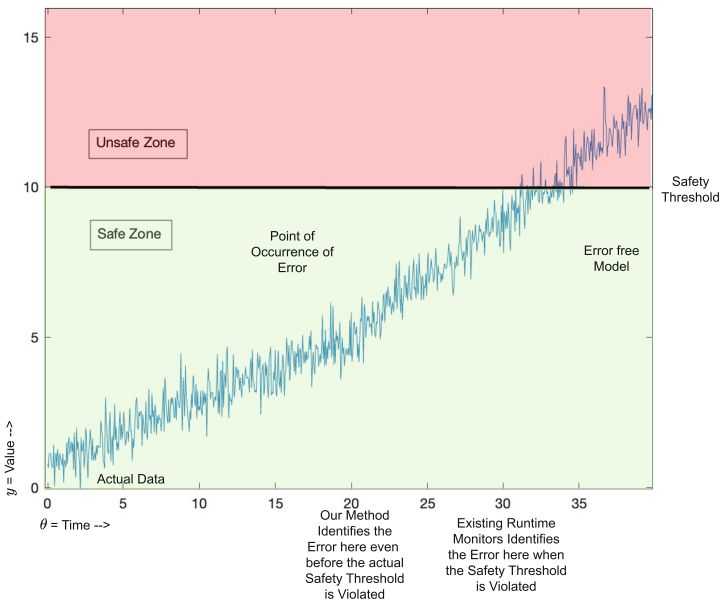


**Fig. 2.** Overview of the Proposed Approach: The diagram illustrates the two-stage process of our methodology. The physics-guided surrogate models facilitate the determination of a conformal range for the surrogate model coefficients. Subsequently, in the Operational Phase, another physics-guided model is learned using real-time operational traces. To ensure the model’s conformance, the critical assessment in this phase involves verifying whether the coefficients of this operational model are within the conformal range identified during the training phase

without the need for predefined error signatures and contributing to the overall safety of Human-Centered Systems. State-of-the-art error detection uses runtime monitoring and involves learning an operational monitor and testing the conformance of the operational data with the monitor’s predictions [20]. An unsafe deviation from the monitor predictions is specified using metric logic such as Signal Temporal Logic (STL) [15, 16]. The satisfaction of the STL is checked by repeatedly evaluating a robustness value on the operational data [15]. We illustrate the inadequacy of such an approach in error detection using a toy example shown in Fig. 3 where there is an unknown error at 20. State-of-art runtime monitoring technique using conformal inference on operational data [15], when implemented in the above example (Fig. 3), is not able to detect the error when it occurs but detects it at a far later point (at 30.1) when the error has already precipitated into a safety violation. In contrast, implementing our strategy as detailed in this paper, annotates the input segment starting at 20 as unsafe. In our approach, we combine continuous model learning and conformal inference on model coefficients to partition the input space into safe and unsafe regions based on whether the learned model is violating the safety STL on model coefficients. Note that other runtime monitoring or error detection techniques that require the predefined error signatures where even unable to detect the presence of the error as the error is assumed to be an unknown-unknown error [11] and such error signatures are not available.

Our solution, and core contribution, is the introduction of continuous model learning and conformal inference on model coefficient. Model coefficients represent the relationship between the input and the output trajectories of the system guided by the physics laws. If an unknown-unknown error affects the system it will lead to inaccurate or deviating model coefficients. This is because the model encapsulates the relationship between the input and the output trajectories and if there is an error it would lead to different model coefficients to compensate for the changes in the system. So in this paper, we propose a model conference on model coefficients rather than on the output trajectories. We show that by converting the STL on model coefficients it is able to detect unknown errors in Human-Centered systems without the need for predefined error signatures.

Our approach is a two-stage process (see Fig. 2), 1) In **Training Stage** - we learn physics-guided surrogate models to determine a conformal range for safe operation on the model coefficients and 2) In **Operational Phase** - we relearn the physics guided surrogate model and check conformance of the model coefficients to determine the existence of errors in the operational traces. Through a series of real-world error detection experiments, we show that a) our method can detect errors even when error signatures are unavailable, b) the technique



**Fig. 3.** The figure illustrates a comparative analysis between current runtime monitors and our approach to error detection. While existing techniques can detect errors at 30 when the safety threshold is breached, our approach can identify errors at 20, precisely when they occur. In this example, the input to the system  $\theta$  is time and  $y$  is the output of the system

is model agnostic that is it doesn't depend on the specific system model of the human-centered system and, c) enables early detection of unknown errors whereby enabling safe operations of these systems.

## 1.1 Contributions

In this paper, we make the following contributions:

- Provide a generic framework for stochastic model conformance checking on model coefficients and not on output trajectories.
- Provide a mechanism to mine physics-guided operational models from operational traces of Human-Centered Black Box Systems.
- Show detection of errors in the artificial pancreas, autonomous vehicles, and aircraft examples.

## 1.2 Paper Organization

The rest of the paper is organized as follows. Section 2 defines the required preliminaries and background work. Section 3 explains the methodology for mining the model coefficients. Section 4 explains how model conformance can be utilized on the model coefficients derived from Sect. 3. Section 5 discusses the case studies we use to verify the proposed method. Section 6 explains the evaluation criteria and Sect. 7 shows the results of the analysis performed on the examples defined in Sect. 5.

## 2 Preliminaries

**Physics Model:** A physics model is a dynamical system expressed using a system of linear time-invariant ordinary differential equations in Eq. 1. The system has  $n$  variables  $x_i$ ,  $i \in \{1 \dots n\}$  in an  $n \times 1$  vector  $\mathcal{X}$ ,  $\mathcal{A}$  is an  $n \times n$  coefficient matrix,  $\mathcal{B}$  is an  $n \times n$  diagonal coefficient matrix.

$$\dot{X}(t) = \mathcal{A}X(t) + \mathcal{B}U(t), Y(t) = \beta X(t) \quad (1)$$

where  $U(t)$  is a  $n \times 1$  vector of external inputs.  $Y(t)$  is the  $n \times 1$  output vector of the system of equations. An  $n \times n$  diagonal matrix,  $\beta$  of 1s and 0s, where  $\beta_{ii} = 1$  indicates that the variable  $x_i$  is an observable output else it is hidden and is not available for sensing.

A formal object  $\hat{\mu}$  is a physics model when the set of models  $\mu$  can be described using the coefficient  $\omega = \mathcal{A} \cup \mathcal{B}$ . The formal object can then take any  $\theta$  as input and given the model coefficients  $\omega$ , generate a trace  $\zeta_\theta = \hat{\mu}(\omega, \theta)$ .

**Trajectory and Models:** A trajectory  $\zeta$  is a function from a set  $[0, T]$  for some  $T \in \mathcal{R}^{\geq 0}$  denoting time to a compact set of values  $\in \mathcal{R}$ . The value of a trajectory at time  $t$  is denoted as  $\zeta(t)$ . Each trajectory is the output of a model  $M$ . A model  $M$  is a function that maps a  $k$  dimensional input  $\theta$  from the input space  $\Theta \subset \mathcal{R}^k$  to an output trajectory  $\zeta_\theta$ .

The input  $\theta \in \Theta$  is a random variable that follows a distribution  $\mathcal{D}_\Theta$ . The model  $M$ , can be simulated for input  $\theta$  and a finite sequence of time  $t_0 \dots t_n$  with  $n$  time steps and generate the trajectory  $\zeta_\theta$  such that  $\zeta_\theta(t_i) = \Sigma(\theta, t_i)$ .

**Trace:** Concatenation of  $p$  output trajectories over time  $\zeta_{\theta_1} \zeta_{\theta_2} \dots \zeta_{\theta_p}$  is a trace  $\mathcal{T}$ .

**Continuous Model Mining:** Given a trace  $\mathcal{T}$ , continuous model mining maps the trace into a sequence  $\Omega$  of  $p$ ,  $\omega_i$ s such that  $\forall i \text{ dist}(\hat{\mu}(\omega_i, \theta_i), \zeta_{\theta_i}) < v$ , where  $\text{dist}(\cdot)$  is a distance metric between trajectories and  $v \approx 0$  is decided by the user.

## 2.1 Signal Temporal Logic

Signal temporal logic are formulas defined over trace  $\mathcal{T}$  of the form  $f(\Omega) \geq c$  or  $f(\Omega) \leq c$ . Here  $f : \mathcal{R}^p \rightarrow \mathcal{R}$  is a real-valued function and  $c \in \mathcal{R}$ . STL supports operations as shown in Eqn. 2.

$$\phi, \psi := \text{true} \mid f(\Omega) \geq c \mid f(\Omega) \leq c \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid F_I\phi \mid G_I\phi \mid \phi U_I\psi, \quad (2)$$

where  $I$  is a time interval, and  $F_I$ ,  $G_I$ , and  $U_I$  are eventually, globally, and until operations and are used according to the standard definitions [6, 9]. To compute a degree of satisfaction of the STL we consider the robustness metric. The **robustness value**  $\rho$  maps an STL  $\phi$ , the trajectory  $\zeta$ , and a time  $t \in [0, T]$  to a real value. An example robustness  $\rho$  for the STL  $\phi : f(\Omega) \geq c$  is  $\rho(f(\Omega) \geq c, \Omega, t) = f(\Omega(t)) - c$ .

## 2.2 Physics-Driven Surrogate Model

A surrogate model is a quantitative abstraction of the black box model  $M$ . A quantitative abstraction satisfies a given property on the output trajectory of the model. In this paper, this quantitative property is the robustness value of an STL property. With this setting, we define a  $\delta$ -surrogate model  $\hat{\mu}$ .

**$(\delta, \epsilon)$  Probabilistic Surrogate model:** Let  $\zeta_\theta$  be a trajectory obtained by simulating  $M$  with input  $\theta$ . Let  $\omega^T$  be the coefficients of the physics-guided representation of the original model. Given a user-specified  $\epsilon$ , the formal object  $\hat{\mu}(\omega, \theta)$  is a  $(\delta, \epsilon)$  probabilistic distance preserving surrogate model if

$$\exists \delta \in \mathcal{R}, \epsilon \in [0, 1] : P(|\rho(\phi, \omega^T) - \rho(\phi, \omega)| \leq \delta) \geq 1 - \epsilon. \quad (3)$$

A  $\delta$  surrogate model guarantees that the robustness value evaluated on a physics model coefficient  $\omega$  derived from the trajectory  $\zeta_\theta$  will not be more than  $\delta$  away from the robustness computed on the coefficients of the original model  $M$ .

## 3 Coefficient Mining from Trajectory

**Problem Definition 1.** Given a set of variables  $\mathcal{X}(t)$ , a set of inputs  $U(t)$ , a  $\beta$  vector indicating observability, and a set  $\mathcal{T}$  of traces such that  $\forall i : \beta_i = 1 \exists T(x_i) \in \mathcal{T}$  and  $\forall u_j(t) \in U(t) \exists T(u_j) \in \mathcal{T}$ .

**Derive:** approximate coefficients  $\mathcal{A}^a$  and  $\mathcal{B}^a$  such that:



**Algorithm 1.** RNN induction algorithm

---

```

1:  $\forall x_i \in X$  create an RNN node with  $n + 1$  inputs and  $x_i$  as the hidden output.
2: for each RNN node corresponding to  $x_i$  do
3:   for each  $j \in 1 \dots n$  do
4:     if  $a_{ij} \neq 0$  then
5:       Add a connection from the output of RNN node for  $x_j$  to the input of
       RNN node for  $x_i$ .
6:     end if
7:   end for
8:   Remove all other inputs in the RNN which does not have any connection.
9:   for each  $j \in 1 \dots n$  do
10:    if  $b_{ij} \neq 0$  then
11:      Add  $u_j$  as an external input to the RNN node for  $x_i$ .
12:    end if
13:  end for
14: end for
15: Assign arbitrary weights to each link.

```

---

- $\forall i, j \ |\mathcal{A}^a(i, j) - \mathcal{A}(i, j)| < \xi$
- $\forall i \ |\mathcal{B}^a(i, i) - \mathcal{B}(i, i)| < \xi$
- Let  $\mathcal{T}^a$  be the set of traces that include variables derived from the solution to differential equation  $\frac{dX(t)}{dt} = \mathcal{A}^a X(t) + \mathcal{B}^a U(t)$  then  $\forall i : \theta_i = 1$ , and  $\forall k \in \{1 \dots N\}, |T^a(x_i)[k] - T(x_i)[k]| < \Psi T(x_i)[k]$ ,

where  $\xi$  is the error in the coefficient estimator, while  $\Psi$  is the error factor for replicating the traces of variables with the estimated coefficients.

### 3.1 Dynamics Induced RNN

For each variable  $x_i \in X$  the system of dynamical equations takes the form in Eq. 4.

$$\frac{dx_i}{dt} = \sum_{j=1}^n a_{ij}x_j + b_{ii}u_i. \quad (4)$$

The RNN induced by the system of equations (Eq. 1) follows Algorithm 1. We explain Algorithm 1 using the linearized Bergman Minimal Model (BMM) as an example. The model is a dynamical system that mimics the glucose-insulin biochemical dynamics in the human body. The Bergman Minimal model is linearized using Taylor Series expansion starting from overnight glucose dynamics and going up to time  $N$ . The linearized model is represented in Eqs. 5, 6 and 7.

$$\frac{d\delta i(t)}{dt} = -n\delta i(t) + p_4 u_1(t) \quad (5)$$

$$\frac{d\delta i_s(t)}{dt} = -p_1 \delta i_s(t) + p_2 (\delta i(t) - i_b) \quad (6)$$

$$\frac{d\delta G(t)}{dt} = -\delta i_s(t)G_b - p_3 (\delta G(t)) + u_2(t)/VoI, \quad (7)$$

The input vector  $U(t)$  consists of the overnight basal insulin level  $i_1b$  and the glucose appearance rate in the body  $u_2$ . The output vector  $Y(t)$  comprises the blood insulin level  $i$ , the interstitial insulin level  $i_s$ , and the blood glucose level  $G$ . For this example, we consider that only the blood glucose level  $G$  is an observable output of the system of equations.  $i_s$  and  $i$  are intermediate outputs that are not measurable for the system of equations and only contribute to the final glucose output.  $p_1, p_2, p_3, p_4, n$ , and  $1/V_oI$  are all the coefficients of the set of differential equations. The resulting DiH-RNN for the BMM using Algorithm 1 is shown in Fig. 4

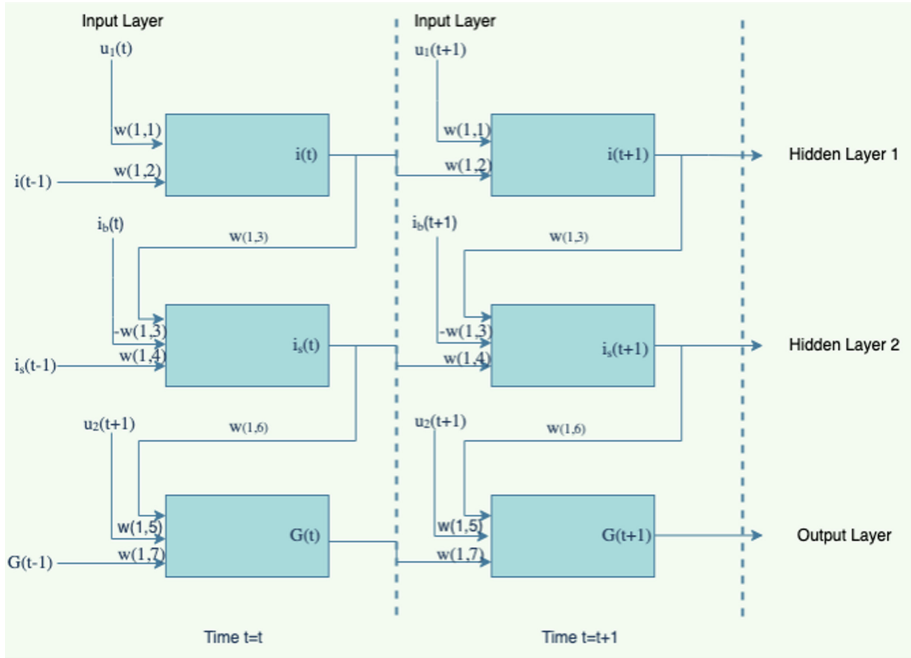


Fig. 4. DiHRNN structure of the Bergman Minimal Model

### 3.2 Forward Pass in DiH-RNN

We prove that the Forward pass on an RNN node estimates the solution of Eq. 4 with error factor  $\Psi$  if  $\tau \leq \min_i \frac{\sqrt{2\Psi}}{a_{ii}}$ . The poof is attached in the appendix.

### 3.3 Backpropagation to Learn Coefficients

The main aim of backpropagation is to derive the approximate coefficient matrices  $\mathcal{A}^a$  and  $\mathcal{B}^a$ . Given an error ratio of  $\phi$ , we have established that the forward

pass is convergent and estimation error is proportional to  $\phi$  if  $\tau \leq \frac{\sqrt{2\Psi}}{|a_{ii}|} \forall i$ . However, we do not know  $a_{ii}$  and hence setting  $\tau$  is a difficult task. Often  $\tau$  is limited by the sampling frequency of the sensor. In this paper, we assume that the  $\tau$  satisfies the condition for convergence of the forward pass. Proposition 1 in [21] shows that for shallow DNNs if all the weights are nonnegative and the activation function is convex and non-decreasing then the overall loss is convex. In such a scenario there exists only a single minima and the gradient descent mechanism is guaranteed to find it.

## 4 Conformal Inference

Conformal Inference is a framework to predict the accuracy of the predictions in a regression framework Conformal Inference is rigorously studied in the following works [11, 15, 16]. We use that basic framework of conformal inference and extend it to model coefficients.

In this approach (Fig. 2), we use error-free operational traces to learn the confidence threshold  $d$ . The process to determine this threshold on model coefficients involves several steps:

- 1) Split the error-free data into training and testing sets.
- 2) From the training set, calculate a set of PGSM model coefficients,  $\omega_e$ .
- 3) For each subset in the testing set, compute model coefficients,  $\omega_i$ , where  $i$  represents the specific subset.
- 4) Using the  $\omega_e$  from the train set we calculate the robustness residue of each test  $\omega_i$ . We define robustness as a quantification of the difference in model coefficient values. For this paper, we consider the maximum deviation of the model coefficients which is explained by the following Eq. 8. Other metrics like minimum deviations and average deviations could have been used but such investigations are beyond the scope of this paper and are left for future investigation.

$$\rho(\phi, \omega) = \max_{j \in \{1 \dots n\}} \text{abs}((\omega[j] - \omega_e[j]) / \omega_e[j]) - \alpha, \quad (8)$$

where  $n$  is the total number of model coefficients in  $\omega_e$ .

- 5) Sort the calculated robustness values in ascending order and identify the residue corresponding to the position defined by  $\lceil (n/2 + 1)(1 - \alpha) \rceil$ .
- 6) The robustness residual value at the given position gives us the confidence range  $d$ , and with it, we derive the confidence interval  $[\rho(\omega) - d, \rho(\omega) + d]$ .

Any new data with unknown errors should result in model coefficients such that the STL robustness residue is beyond the range  $[\rho(\omega) - d, \rho(\omega) + d]$ .

## 5 Case Studies

Human-centered systems are those where failure could result in catastrophic outcomes, such as loss of life, significant property damage, or harm to the human

**Table 1.** Physical model coefficients derived using DiHRNN for train and test set

Train/Test	$p_1$ 1/min	$p_2$ 1/min	$p_3 \frac{10^{-6}}{\mu U.min^2}$	$p_4$	$n$ 1/min	$VoI$ dl	$G_b$ mg/dl	Residue
Simulation Set-tings	0.098	0.035	0.028	0.05	0.1406	199.6	-80	NA
Train	0.0978	0.0349	0.0262	0.0508	0.1406	198.134	-80.64	0
Test 1	0.0982	0.0329	0.0256	0.0530	0.1405	198.1340	-80.2774	0.0225
2	0.0979	0.0332	0.0274	0.0533	0.1407	198.1340	-85.0589	0.0028
3	0.0980	0.0348	0.0262	0.0528	0.1405	198.1340	-85.0973	0.0011
4	0.0981	0.0343	0.0267	0.0515	0.1405	198.1340	-80.6921	-0.0168
5	0.0979	0.0317	0.0273	0.0548	0.1407	198.1340	-82.7676	0.0328
6	0.0980	0.0328	0.0275	0.0534	0.1404	198.1340	-82.3447	0.0048

participant. In this section, we present three real-world safety critical examples. Each example features a human integrated into the operational dynamics, as outlined by the architecture depicted in Fig. 1. The inclusion of humans within the operational framework elevates the criticality of these systems, significantly increasing the risk of harm. In these cases, the problem of detection of unknown errors is even more important.

### 5.1 Automated Insulin Delivery System Example

In the Automated Insulin Delivery (AID) system, the glucose-insulin dynamics is given by the Bergman Minimal Model (BMM) represented as 5, 6, and 7 and is explained in detail in Sect. 3.1. For this paper, we consider the unknown error of insulin cartridge error in the automated insulin delivery system. The error signature of the error was unavailable at the time of the error as this error was never seen before. The human being part of the system being controlled made measuring the effects even more complicated. While the controller operated under the assumption of flawless insulin administration, the actual delivery to the human body (the system) was compromised, leading to a significant disparity between the system’s state as perceived by the controller and its true state (Table 1).

### 5.2 Aircraft Example

Pitch control in an aircraft is automated using a Proportional Integrative Derivative (PID) Controller. The pitch control system considers a linear system model described by Eq. 9 [14].

$$\begin{aligned}
 \dot{x}_\alpha &= c_{\alpha\alpha}x_\alpha + c_{\alpha q}x_q + c_{\alpha\delta}u_\delta, \quad \dot{x}_q = c_{q\alpha}x_\alpha - c_{qq}x_q + c_{q\delta}u_\delta \\
 \dot{x}_\theta &= c_{\theta\theta}x_\theta, \quad y(t) = x_\theta.
 \end{aligned}
 \tag{9}$$

Here  $x_\alpha$  is the angle of attack (AoA),  $x_q$  is the pitch rate,  $u_\delta$  is the elevator angle, and  $x_\theta$  is the pitch angle of the aircraft. The controller is a PID and based on a pitch angle set point derives the elevator angle  $u_\delta$ . Hence,  $u_\delta$  is the input to the aircraft dynamics, while  $x_\theta$  is the output of the dynamical model. A trajectory is the continuous time value of state variables in between two elevator angle inputs from the PID.

For this example, we consider the unknown MCAS error that caused the accidents in the Boeing 787 aircraft. The cause was also unknown due to the black box abstraction of the MCAS system. The human participant (the pilot in this case) did not know that the faulty AOA sensor was being used to control the plant.

### 5.3 Autonomous Driving Example

An autonomous car detects another static car in its lane and attempts to stop before crashing into the car ahead. The kinematics of the car is given by:

$$\begin{aligned} \dot{a}_x &= -0.01s_x + 0.737 - 0.3v_x - 0.5a_x, \\ \dot{v}_x &= 0.1a_x, \quad \dot{s}_x = v_x - 2.5. \end{aligned} \tag{10}$$

For this example, we consider the unknown error of a zero-day vulnerability in the controller code. The vulnerability caused the black box controller code to change from  $f_c$  to  $f'_c$  in Fig. 1. Originating from a zero-day vulnerability, the full impact on the system was uncertain, given that this vulnerability had not been detected before.

## 6 Evaluation Method and Metrics

Human-centered systems are safety-critical, and it is necessary to identify unknown errors to shield the human participant from harm. The performance of zero-shot detection of unknown errors is quantified in terms of the true positive rate of the detection algorithm. We designate the approach as Detected (D) if it can identify the Unknown-Unknowns, and Undetected (ND) if it cannot. The availability of real data for such real-world safety-critical systems with unknown errors is fairly limited. So, here we use simulators developed in MATLAB to generate data for such unknown errors in real-world complex systems.

### 6.1 Unknown-Unknown Scenario Simulation

For the AID example, we use the shunted insulin model to generate the traces with the insulin cartridge errors. We vary the amount of insulin blockade percent between 20 to 80% and the time until insulin release from 50 to 150 mins. The scenarios generated for the insulin cartridge problem are listed in Table 2.

For the AoA error in the MCAS system, we use any error or noise rate of 20–25% in the AoA measurement and use that to derive the coefficients at the model of the pitch control system.

For the autonomous vehicle example, an integer overflow vulnerability in the control software is considered where instead of declaring  $Q$  as an `uint16_t` variable it is mistakenly defined as `int8_t`. This means that instead of setting  $Q(1, 1) = 10,000$ , it is now set at  $Q(1, 1) = 16$ . This can potentially cause a crash since the controller is less aggressive.

## 6.2 Baseline Strategy

We replicate the model conformance-based strategy described in [15, 16] to the best of our knowledge. In the work, the authors learn a surrogate model of the system under test and use it to find the robustness range of the output values. During operation, a new model is learned from the test traces and checked if the robustness values lie within the robustness range. If the robustness value of the test system is outside the range then the system under test is termed to have deviated from the approved characteristics.

## 7 Results

In safety-critical systems that involve human participants, it is of utmost necessity to detect every possible error to stop the faulty system from causing any harm to humans. It is established that detecting such errors that is the number of true positives detected is far more important than other metrics. So, in this paper, we consider the true positive rate of the different detection algorithms.

**Implementation:** The Unknown-Unknown (U2) detection mechanism and DiHRNN were implemented on a single thread of an Intel i7 CPU processor, without parallelism or optimizations. Initial results indicate that DiHRNN is highly suitable for model learning on edge devices, as it requires significantly less memory and computational power compared to traditional deep learning methods [13]. On average, DiHRNN takes approximately 12 seconds to relearn each model and is also highly memory-efficient. For instance, in the AID example, DiHRNN requires only about 9 bytes of memory to store the model weights, as it needs to store just 9 weights without any optimizer states or activation functions.

### 7.1 Automated Insulin Delivery System Example

Table 2 shows that for the insulin cartridge problem, the model conformance results show that the robustness values under various input configurations are falling outside the range. Hence, these scenarios are deemed to be non-conformal to the original model. Using the technique defined in this paper, it was able to detect all the unknown errors simulated for evaluation purposes without the need for error signatures and have a positive predictive value of 100%.

**Table 2.** Comparison of physical model coefficients derived using DiH-RNN for different Insulin Blockages to detect the errors, D in the robustness column means error detected and Robustness is beyond  $[-0.0216, 0.0376]$ . Insulin = 7.5 U, Meal = 20 g

Insulin Block Percentage	Time until insulin release	$p_1$ 1/min	$p_2$ 1/min	$p_3 \frac{10^{-6}}{\mu U \cdot min^2}$	$p_4$	$n$ 1/min	$VoI$ dl	$G_b$ mg/dl	Robustness	Model Conformance on coefficients (Our Method)	Model Conformance on Output [15, 16]
20	150	0.098	0.033	0.018	0.065	0.1404	268.55	-51.46	0.37	(D)	ND
40	120	0.098	0.034	0.018	0.053	0.1402	287.92	-68.32	0.3885	(D)	ND
80	90	0.098	0.034	0.019	0.068	0.1401	235.25	-58.68	0.36	(D)	ND
70	70	0.098	0.033	0.020	0.068	0.1400	216.14	-48.12	0.43	(D)	ND
60	50	0.098	0.034	0.019	0.068	0.1405	180.48	-69.76	0.35	(D)	ND
Phantom 20	150	0.098	0.0269	0.0194	0.058	0.1402	155.89	-54.104	0.32	(D)	ND
Phantom 40	120	0.098	0.0339	0.0218	0.0579	0.1402	307.06	-60.73	0.5284	(D)	ND
Phantom 80	90	0.098	0.0344	0.0217	0.0503	0.1401	143.43	-64	0.27	(D)	ND
Phantom 70	70	0.098	0.0348	0.0229	0.0655	0.139	169.20	-48.26	0.48	(D)	ND
Phantom 60	50	0.0983	0.0349	0.0187	0.0554	0.1400	317.86	-55.12	0.5825	(D)	ND

### 7.2 Aircraft Example

As shown in Table 3, the model conformance with STL on the model outputs failed to recognize errors as the outputs fell within the defined safe and robust range. In contrast, our proposed detection technique by applying model conformance to the model’s parameters, successfully identified 9 out of 10 such errors immediately upon their occurrence and had a positive predictive value of 90%.

**Table 3.** Comparison of physical model coefficients derived using DiH-RNN for different AoA errors and error timings to detect errors, D in the robustness column means error detected and Robustness is beyond  $[0.0299, 0.1116]$

Set point SP (rads)	SP change time (s)	AoA error (rad)	Error Time (s)	$c_{\alpha\alpha}$ 1/s	$c_{\alpha q}$	$c_{\alpha\delta}$ 1/s	$c_{q\alpha}$ 1/s	$c_{qq}$ 1/s	$c_{q\delta}$ 1/s <sup>2</sup>	$c_{\alpha q}$ 1/s	Robustness	Model Conformance on coefficients (Our Method)	Model Conformance on Output [15, 16]
0.2	0	0.6	5	-0.276	53.7	0.24	-0.0118	-0.475	0.0232	60.1	0.136	(D)	ND
0.5	5	0.2	7	-0.258	47.6	0.24	-0.0123	-0.482	0.0205	62.9	0.156	(D)	ND
0.4	2	0.4	10	-0.282	45.5	0.24	-0.0115	-0.51	0.0213	66.11	0.22	(D)	ND
0.8	5	0.4	5	-0.281	60.2	0.27	-0.0126	-0.4	0.0219	65.2	0.13	(D)	ND
0.1	5	0.6	5	-0.269	52.4	0.25	-0.0129	-0.489	0.0231	63.2	0.17	(D)	ND
0.1	7	0.6	5	-0.28	63.6	0.26	-0.0123	-0.367	0.0214	60.23	0.11	(ND)	ND
0.1	9	0.7	2	-0.257	65.0	0.27	-0.0136	-0.354	0.0219	58.3	0.16	(D)	ND
0.4	9	0.9	2	-0.26	64.8	0.25	-0.0138	-0.398	0.0243	61.2	0.17	(D)	ND
0.1	10	0.6	10	-0.293	67.5	0.27	-0.0136	-0.372	0.023	66	0.17	(D)	ND
0.3	1	0.3	10	-0.302	46.1	0.28	-0.0125	-0.46	0.0214	63.3	0.18	(D)	ND

### 7.3 Autonomous Driving Example

We conducted 11 simulations of the autonomous braking system, using the data to train a deep learning model for assessing the reliability of the system’s output. Subsequently, we carried out an additional 11 simulations introducing braking errors. The vulnerable controller code was executed to obtain the traces starting from the same initial  $s_x$  and  $v_x$  as training. The average robustness residue is  $-17.395 (\pm 2.1)$ , with all vulnerable traces falling outside the robustness range. Our proposed method using DiH-RNN, implementing STL on the model’s parameters, detected all 11 errors and had a positive predictive value of 100%.

## 8 Future Works

In this paper, we present DiHRNN as the primary mechanism for real-time model re-learning. However, further investigation is needed to evaluate its performance compared to other model learning techniques. Currently, the framework operates on a single-thread CPU, and future optimizations are required to enhance its speed. The latency for detecting the first unknown-unknown within the framework is approximately 13 seconds. While this latency is acceptable for many applications, future improvements are necessary to reduce it, making the framework suitable for real-time analysis in applications that demand lower latency.

## 9 Conclusions

This paper proposed a model-agnostic framework for the detection of unknown errors in operational human-centered systems without the need for error signatures. By employing a physics-guided surrogate model to track the physical system’s behavior and using a hybrid RNN approach to derive model coefficients, our method identifies deviations using conformal inference techniques, signaling unknown errors in the operational system. With our technique, we can detect errors that haven’t been identified before and can stop the system from causing harm to the human participant. Our results demonstrate that this method surpasses existing state-of-the-art error detection techniques in identifying errors without relying on pre-established error definitions. However, it’s important to note that our method’s efficacy relies heavily on the training and testing data, particularly when determining the model coefficients. Furthermore, the impact of available training data on error detection accuracy needs further investigation.

**Acknowledgments.** This work is partly funded by the DARPA AMP grant (N6600120C4020) and the DARPA FIRE grant (P000050426). Opinions, interpretations, conclusions, and recommendations in the paper are those of the authors and not endorsed by the agency.



## References

1. Agha, G., Palmiskog, K.: A survey of statistical model checking. *ACM Trans. Model. Comput. Simul. (TOMACS)* **28**(1), 1–39 (2018)
2. Banerjee, A., Kamboj, P., Maity, A., Salian, R., Gupta, S.: High fidelity fast simulation of human in the loop human in the plant (hil-hip) systems. In: *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pp. 199–203 (2023)
3. Banerjee, A., Lamrani, I., Gupta, S.K.: Faultex: explaining operational changes in terms of design variables in cps control code. In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 485–490. IEEE (2021)
4. Banerjee, A., Maity, A., Gupta, S.K., Lamrani, I.: Statistical conformance checking of aviation cyber-physical systems by mining physics guided models. In: *2023 IEEE Aerospace Conference*, pp. 1–8. IEEE (2023)
5. Banerjee, A., Maity, A., Kamboj, P., Gupta, S.K.: CPS-LLM: large language model based safe usage plan generator for human-in-the-loop human-in-the-plant cyber-physical system. *arXiv preprint [arXiv:2405.11458](https://arxiv.org/abs/2405.11458)* (2024)
6. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) *FORMATS 2010*. LNCS, vol. 6246, pp. 92–106. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-15297-9\\_9](https://doi.org/10.1007/978-3-642-15297-9_9)
7. Dreossi, T., et al.: VERIFAI: a toolkit for the formal design and analysis of artificial intelligence-based systems. In: Dillig, I., Tasiran, S. (eds.) *CAV 2019*. LNCS, vol. 11561, pp. 432–442. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-25540-4\\_25](https://doi.org/10.1007/978-3-030-25540-4_25)
8. Dutta, S., Jha, S., Sankaranarayanan, S., Tiwari, A.: Learning and verification of feedback control systems using feedforward neural networks. *IFAC-PapersOnLine* **51**(16), 151–156 (2018)
9. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. *Theoret. Comput. Sci.* **410**(42), 4262–4291 (2009)
10. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: Majumdar, R., Kunčák, V. (eds.) *CAV 2017*. LNCS, vol. 10426, pp. 3–29. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-63387-9\\_1](https://doi.org/10.1007/978-3-319-63387-9_1)
11. Maity, A., Banerjee, A., Gupta, S.: Detection of unknown-unknowns in cyber-physical systems using statistical conformance with physics guided process models. *arXiv preprint [arXiv:2309.02603](https://arxiv.org/abs/2309.02603)* (2023)
12. Maity, A., Banerjee, A., Lamrani, I., Gupta, S.K.: Cyphytest: cyber physical interaction aware test case generation to identify operational changes. In: *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 01–06. IEEE (2022)
13. Maity, A., Banerjee, A., Lamrani, I., Gupta, S.K.: Context aware model learning in cyber physical systems. In: *2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 1–6. IEEE (2024)
14. Messner, B., Tilbury, D., Hill, R., Taylor, J.: Control tutorials for matlab and simulink: Aircraft pitch. Retrieved from <https://web.archive.org/web/20200509164711/http://ctms.engin.umich.edu/CTMS/index.php> (2020)
15. Qin, X., Xia, Y., Zutshi, A., Fan, C., Deshmukh, J.V.: Statistical verification using surrogate models and conformal inference and a comparison with risk-aware verification. *ACM Trans. Cyber-Phys. Syst.*

16. Qin, X., Xian, Y., Zutshi, A., Fan, C., Deshmukh, J.V.: Statistical verification of cyber-physical systems using surrogate models and conformal inference. In: 2022 ACM/IEEE 13th international conference on cyber-physical systems (ICCPS), pp.116–126. IEEE (2022)
17. Safikhani, A., Shojaie, A.: Joint structural break detection and parameter estimation in high-dimensional nonstationary VAR models. *J. Am. Stat. Assoc.* **117**(537), 251–264 (2022)
18. Sun, X., Khedr, H., Shoukry, Y.: Formal verification of neural network controlled autonomous systems. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, pp. 147–156 (2019)
19. Tran, H.-D., et al.: Star-based reachability analysis of deep neural networks. In: ter Beek, M.H., McIver, A., Oliveira, J.N. (eds.) FM 2019. LNCS, vol. 11800, pp. 670–686. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-30942-8\\_39](https://doi.org/10.1007/978-3-030-30942-8_39)
20. Weaver, K.W., Hirsch, I.B.: The hybrid closed-loop system: evolution and practical applications. *Diabetes Technol. Therapeut.* **20**(S2), S2-16 (2018)
21. Yuan, J., Weng, Y.: Physics interpretable shallow-deep neural networks for physical system identification with unobservability. In: 2021 IEEE International Conference on Data Mining (ICDM), pp. 847–856. IEEE (2021)



# Source-Free Test-Time Adaptation For Online Surface-Defect Detection

Yiran Song<sup>1</sup>, Qianyu Zhou<sup>1</sup>, and Lizhuang Ma<sup>1</sup>✉

Shanghai Jiao Tong University, Shanghai, China  
{songyiran,zhouqianyu,lzma}@sjtu.edu.cn

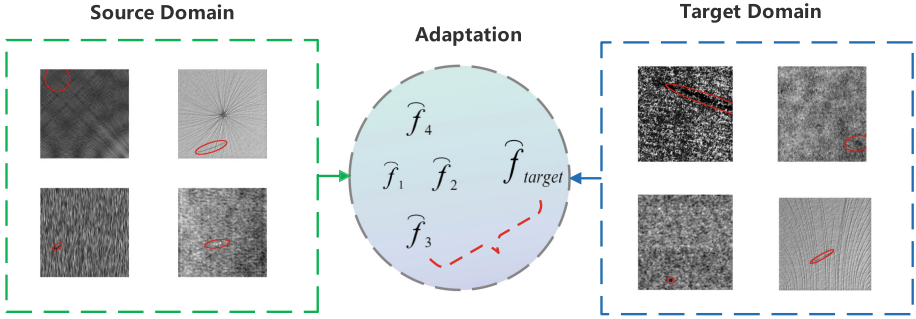
**Abstract.** Surface defect detection is significant in industrial production. However, detecting defects with varying textures and anomaly classes during the test time is challenging. This arises due to the differences in data distributions between source and target domains. Collecting and annotating new data from the target domain and retraining the model is time-consuming and costly. In this paper, we propose a novel test-time adaptation surface-defect detection approach that adapts pre-trained models to new domains and classes during inference. Our approach involves two core ideas. Firstly, we introduce a supervisor to filter samples and select only those with high confidence to update the model. This ensures that the model is not excessively biased by incorrect data. Secondly, we propose the augmented mean prediction to generate robust pseudo labels and a dynamically-balancing loss to facilitate the model in effectively integrating classification and segmentation results to improve surface-defect detection accuracy. Our approach is real-time and does not require additional offline retraining. Experiments demonstrate it outperforms state-of-the-art techniques.

**Keywords:** Surface-defect detection · Test-time adaptation · Source-free domain adaptation · Online adaptation

## 1 Introduction

With the advent of deep learning [7–9, 14, 15, 30, 32, 48, 55, 60, 61, 65], surface-defect detection (SDD) [58] has made great progress recently in industrial scenarios. Unfortunately, gathering and labeling anomalous samples is costly. The collected datasets of industrial production are usually limited, which hinders effective training. As a result, models excel under the same training distribution but suffer from accuracy degradation due to domain shifts, *e.g.*, varying textures, and new defect classes, which usually appear in testing.

Test-time Adaptation (TTA) is a task that uses unsupervised testing data to infer the target domain distribution. The online, unlabeled data arrives continuously, demanding immediate model updates and decisions. Various TTA networks have been proposed, such as TENT [47] and CoTTA [49]. These methods



**Fig. 1.** Visualization of the domain discrepancy in cross-domain surface-defect detection.  $f$  represents the optimal parameters that can be learned. Our goal is to find a path that can span the difference between the source and target domains

enable models to adapt to different data distributions during the test time. However, directly applying TTA methods to industrial scenarios will encounter several challenges. Firstly, Table 1 shows that the dataset sizes of industrial datasets are usually significantly smaller than classical datasets. A small dataset can lead to a higher likelihood of encountering untrained knowledge and lead the unstable performance during inference. Besides, different from existing TTA that usually assumes that the source domain and the target domain share the same label space, a more specific challenge in industrial scenarios is that it will encounter novel classes of defects during the online adaptation (Fig. 1).

**Table 1. Comparison of classical test-time adaptation dataset and industrial dataset** We can see that for the same TTA task, the dataset for the traditional image segmentation tasks is much larger than that for the surface-defect detection tasks

Classical Dataset	Class	Total number	Industrial dataset	class	Total number
CIFAR10-C	15*5	750000	KolektorSDD	1	399
ImageNet-C	15*5	3750000	DAGM	10	8050

Motivated by the above analysis, we present a novel test-time adaptation method for surface-defect detection. To enhance the adaptability toward the target domains, we introduced a supervisor to predict the sample reliability, which is initialized with source domain parameters, and kept constant during the testing. To bolster the pipeline’s robustness, we design two strategies to improve the transferability: augmented mean prediction and dynamically-balancing loss. Concretely, augmented mean prediction generates multiple predictions per sample and combines them for a more stable pseudo-label. Besides, dynamically-balancing loss adjusts the model’s learning focus over time to enhance the robustness of the model. Our contributions are summarized as follows.

**Table 2. The difference between our proposed test-time adaptation and related adaptation settings.** We compared the differences in the related settings. Our approach requires only unlabeled test data. The test domain is allowed to have different classes from the source domain. Our approach is online updates on the test domain without source domain data and offline retraining

setting	source data	target data	new class	train stage	test stage
fine-tuning	no	stationary+labeled	yes	yes	no
standard domain adaptation	yes	stationary	yes	yes	yes
standard test-time training	yes	stationary	yes	yes(aux task)	yes
fully test-time adaptation	no	stationary	no	no(pre-trained)	yes
continual test-time adaptation	no	continually changing	no	no(pre-trained)	yes
our industrial setting	no	continually changing	yes	no(pre-trained)	yes

- We propose a real-time, test-time adaptation method for online surface-defect detection tasks, without offline retraining or source domain data reuse.
- To bolster pipeline robustness, we introduce a supervisor to filter samples, devise augmented mean prediction, and dynamically-balancing loss to generate more stable pseudo-labels and combat catastrophic forgetting.
- Experimental results show our presented approach outperforms existing state-of-the-art methods on various industrial datasets.

## 2 Related Work

*Domain Adaptation.* Our work is related to **unsupervised** domain adaptation(UDA), **source-free** domain adaptation, and **test-time adaptation** (TTA). Though Domain Generalization (DG) methods [21, 33–35, 50, 66–68] can improve the model’s generalizability, they only utilize the seen data in the training stage, which fails in utilizing the information of the target data, thus resulting in unsatisfactory performance on the target domain. In contrast, UDA methods [10, 12, 13, 17, 31, 41, 43, 52, 54, 62–64, 70] aim to adapt a model given unsupervised data, which access labeled data from the source domain and unlabeled data from the target domain at the same time. In our setting, source data is not needed during the adaptation time, and the model is adapted using the unlabeled data solely from the target domain. The source-free domain adaptation methods [24, 27, 29, 42, 69] require no data from the source domain for the adaptation process. However, most of them are deployed in an offline manner and cannot tackle the online streaming data.

*Test-Time Adaptation.* Our work is belong to test-time adaptation [45, 47] category. [25, 27, 39] utilize generative models to acquire feature alignment. Test entropy minimization (TENT) [47] is proposed to adapt the test data by minimizing the prediction entropy. Source hypothesis transfer (SHOT) [29] utilizes both entropy minimization and a diversity regularizer for adaptation. [36] apply

a diversity regularizer combined with an input transformation module to further improve the performance. [22] uses a separate normalization convolutional network to normalize test images. [20] updates the final classification layer during inference time using pseudo-prototypes. [59] proposes a regularized entropy minimization procedure at test-time adaptation, which requires approximating density during training time. [19, 28, 56] Update the statistics in the Batch Normalization layer using the target data. [18, 23] extend test-time adaptation to semantic segmentation.

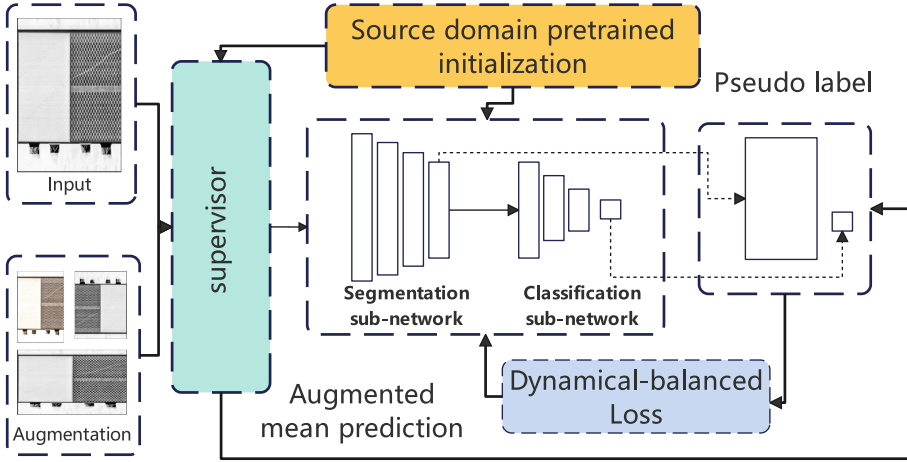
*Unsupervised Learning for Surface-Defect Detection.* Unsupervised learning in industry learns features through reconstruction objective [4], adversarial loss [11] or self-supervised objective [6], without the use of annotated data. Although these methods can significantly reduce the cost of acquiring annotated data, they perform significantly worse compared to fully supervised methods.

### 3 Method

We assume that the model is initialized by parameters  $\theta$  pre-trained on the source domains. Our goal is to adjust the model in an online manner during the test time. The input  $x_t$  is provided at time  $t$  sequentially, drawn from target distribution  $P^t(\mathbf{X}) \neq P^s(\mathbf{X})$ . The parameters of the model  $f_{\theta_{t-1}}$  are updated to  $f_{\theta_t}$  in time  $t$  based on the input  $x_t$ . Our setup is motivated by the need for surface-defect detection in industrial scenarios. In Table 2, we list the differences between our industrial setup and the relevant adaptation setups that already exist to better show the necessity of our work. Our work focuses on source-free, real-time inference while having fewer constraints on the target domain and higher generalization capabilities. Specifically, we allow the test domain to appear as novel anomalous classes and different texture information that do not appear in the source domain, rather than just adding additional noise on the same class [47, 49]. Our setting meets two conditions. 1) Data: only using the unlabeled target domain data. 2) Updating way: the model is updated online and does not require to be retrained offline.

As shown in Fig. 2, our method contains two key parts. Firstly, **we introduce a supervisor as a “gate” to filter the testing data.** We identify untrustworthy data that the model cannot confidently classify, only performing inference on them but excluding them for model updates. For plausible data where the model makes confident predictions, we use them to learn the target data and update the model accordingly. Secondly, we enhance TTA’s robustness with two core modules: **improved average prediction** and **dynamically balancing loss.** We employ the model’s average predictions to reduce outlier impact and boost performance. We also adjust weights in the loss function based on prediction errors to prevent overfitting to training data.

*Base Model.* We use a lightweight two-stage CNN-architecture model [2] as our base model. This model is a two-stage end-to-end structure that supervises both the segmentation and classification results during the training time.



**Fig. 2. Architecture of our proposed method** – We initialize all modules using the parameters trained on the source domain. Each sample on the target domain is fed to the supervisor to get a score, and only reliable samples are used. The augmented samples are fed into the *supervisor* to obtain prediction results, which are combined with the results inferred from the model to generate the pseudo label. We use the pseudo label to update the model with a *dynamically-balancing loss*

The number of parameters is much smaller than common network models (*e.g.*, ResNet, ViT, etc.), meeting the performance requirements of industrial detection (*i.e.*, real-time inference updates).

### 3.1 Supervisor

In the test-time adaptation testing, using untrustworthy predicted results as pseudo-labels for self-supervised learning can lead to model performance bottlenecks. In industrial scenarios, models are more sensitive to such inaccurate pseudo-labels, because the segmentation results for anomaly detection in industrial settings are usually small, and be more sensitive to subtle changes. Additionally, due to performance constraints, industrial detection models are typically smaller, and using erroneous labels for model updates can cause the model to move even further in the wrong direction.

To deal with accumulated errors, we create a supervisor with a structure similar to the model. The key distinction is that the supervisor using parameters from source domain training and **does not** undergo backpropagation. It retains the original knowledge. When the supervisor finds low prediction reliability  $p$  below a set threshold  $p_{th}$ , the adaptive phase is skipped. This is because pseudo-labels from low-confidence predictions can mislead the model. We skip such samples to avoid steering the model in the wrong direction.

Adapting to new distributions can lead to losing knowledge from the source domain, causing severe information loss. Unlike others, we are constrained to

not retrain the model from the source, and prolonged self-training may introduce errors, affecting label accuracy. In our approach, the supervisory module employs pre-trained parameters from the source domain, without further updates in the entire TTA process. It holds source domain knowledge and guides pseudo-label generation, preventing memory loss.

### 3.2 Augmented Mean Prediction

As shown in Sect. 3.1, the supervisor we proposed is initialized with parameters pre-trained on the source domain and is not updated during the test time. It comes with all the knowledge learned from the source domain. We use it to generate pseudo-labels to introduce source domain information. Besides, we propose a method based on augmented average predictions. Specifically, we use sample images that have been data enhanced in many different ways *e.g.*, stretched, cropped, flipped, *etc.*) to input into the supervisor to obtain the prediction results. At the same time, the complete original sample images are also input into the model to obtain  $Y$ . When performing the filtering operation, each sample is given a confidence value  $p$  for the pseudo label. The pseudo label of this image is finally obtained from all the above predictions by weighting averaged. The weight  $w$  is a function related to the confidence level  $p$  value. The lower the confidence level, the more the model will refer to the prediction results the supervisor gave (i.e., source domain knowledge) using the augmented picture. The specific calculation formula is shown below:

$$\tilde{y}_t = \frac{1}{N} \sum_{i=0}^{N-1} f_{\theta_t}(\text{aug}_i(x_t)) \quad (1)$$

where  $p_{\text{aug}_i} > p_{\text{th}}$ . Here,  $p_{\text{aug}_i}$  and  $p_{\text{th}}$  refer to the confidence of the augmented image and the confidence threshold, respectively.

### 3.3 Dynamically-Balancing Loss

Traditional TTA methods compute the loss function based solely on the segmentation or classification results. However, these are not suitable for surface-defect detection. This is because the segmentation portion of anomaly detection datasets is much smaller than the background and the textures are complex, making segmentation quite difficult. In addition, for anomaly detection tasks, correct classification (whether samples with surface-defects can be identified) is of great practical significance. Therefore, during the TTA phase, we simultaneously compute the classification and segmentation losses.

We propose a dynamic weight loss function rather than a fixed weight loss. Specifically, our loss function is defined as:  $L_{\text{total}} = \lambda_{\text{class}} L_{\text{class}} + (1 - \lambda_{\text{class}}) L_{\text{seg}}$ ,  $\lambda_{\text{class}} = 1 - t/N$ , where  $t$  is the current-time index and  $N$  is the number of test dataset. It utilizes a time-dependent weighting scheme to balance the classification and segmentation losses during



the self-adaptive testing phase. The underlying principle is that the model’s performance in different tasks changes as it adapts to the target domain. By giving priority to the classification loss at the beginning, the model can focus on correctly classified samples, which is crucial for identifying anomalous regions. As the model’s distribution shifts towards the target domain, the segmentation loss is given more weight, enabling the model to capture the complex features of the anomalous regions more accurately.

For the specific calculation of these two components, we tried various combinations of common loss functions, including Kullback-Leibler divergence loss, BCE loss, softmax loss, and DICE loss. Through our experiments, we find that the softmax loss combined with Kullback-Leibler divergence loss achieves the best learning effect, specifically defined as follows.

$$L_{\text{class}} = \frac{1}{n} \sum_{i=1}^n \left( -\log \frac{e^{l_{i,Y^{(i)}}}}{\sum_{k=1}^C e^{l_{i,k}}} \right) \quad (2)$$

$$L_{\text{seg}} = -\sum x \log(p) - \left( -\sum x \log(x) \right) \quad (3)$$

To illustrate our algorithm more clearly, the complete process is shown in Algorithm 1. Through the filtering by the supervisor and the optimization, our algorithm effectively reduces error accumulation and catastrophic forgetting when test time adaptation is performed on the target domain.

### 3.4 Model Update Pipeline

To illustrate our algorithm more clearly, the complete process is shown in Algorithm 1. Our algorithm effectively reduces error accumulation and catastrophic forgetting when unsupervised learning is performed on the target domain.

---

#### Algorithm 1 Framework for online test-time adaptation

---

**Initialize:** A model  $f_{\theta_0}(x)$  and Its supervisor  $m_{\theta_0}(x)$  (Both initialized with parameters  $\theta_0$  which obtained by pre-training on the source domain  $D_s$ ). The threshold used for filtering samples  $p_{th}$ .

**Input:** For each time step  $t$ , unlabeled data  $x_t$  sampled from target domain  $D_t$ .

- 1: Input Provide input  $x_t$  to the supervisor  $m_{\theta_0}(x)$  and obtain the confidence probability  $p$ ;
  - 2: **if**  $p > p_{th}$  **then**
  - 3: Provide the set of Augment  $x_t$  to supervisor  $m_{\theta_0}(x)$  and obtain predictions;
  - 4: Provide  $x_t$  to model  $f_{\theta_0}(x)$  and obtain predictions;
  - 5: Use augmented mean prediction method to acquire pseudo-label of  $x_t$
  - 6: Upgrade model  $f_{\theta_0}(x)$  by loss in 3.3
  - 7: **end if**
  - 8: Calculation of prediction result  $y_t$
- Output:** Prediction  $y_t$ , Updated model  $f_{\theta_t}(x)$
-

## 4 Experiments

### 4.1 Datasets and Pre-training

*DAGM 2007 Dataset.* DAGM dataset [51] is a well-known benchmark database for surface-defect detection. It contains images of various surfaces with artificially generated defects. Surfaces and defects are split into 10 classes of various difficulties. We randomly selected four types of samples from the DAGM dataset as the training set for pre-training the model. The model is then no longer exposed to the source domain dataset but is validated on the remaining six unseen anomaly classes. As shown in Fig. 3, there are significant differences in the distribution of the ten anomaly classes.

*KolektorSDD Datasets.* [46] is annotated by the Kolektor Group. The images were captured in a controlled industrial environment in a real-world case. The dataset consists of 399 images, of which 52 images with visible defects and 347 images without any defects. The original width is 500 px, and the height is from 1240 to 1270 px. We resize images to  $512 \times 1408$  for training and evaluation. For each item, the defect is only visible in at least one image, while two items have defects on two images, which means there were 52 images where the defects are visible. The remaining 347 images serve as negative examples with non-defective surfaces. Since KolektorSDD does not have the same subclass division as DAGM, we manually divide the anomalous samples into two parts with large morphological differences and use one part for training while validating the other part to demonstrate its adaptive ability on target domains with different distributions.

*Pre-training.* Following the work in [2], we use a two-stage model as our base model, where the segmentation is performed in the first stage, followed by a per-image classification in the second stage. We train the network using stochastic gradient descent with no weight decay and no momentum. We initialize the base model and the Supervisor using the trained parameters  $\theta_0$ . Since the dataset suffers from severe positive and negative sample imbalance, we use low sampling of negative samples, and in each training epoch, we select negative samples of the same size as the positive subset of the sample. We also ensured that all negative images were used approximately equally often during the learning process. Our pre-training on the source domain does not require additional measures to improve the generalization ability of the model. For the DAGM dataset, we train 60 epochs with a learning rate of 0.01 batch size = 5. For the KolektorSDD dataset, we train 35 epochs, using a learning rate of 0.5. After training with the source domain data, the model does not need to use the training data again in the subsequent stages and will not be retrained offline again.

### 4.2 Results

*Inference Time.* As we have highlighted, our model is a lightweight, online-inference CNN model to meet the requirements of industrial scenarios. Our

proposed method achieves 32 fps on  $512 \times 512$  images (DAGM) and 13 fps on  $512 \times 1408$  images (KolektorSDD). For TTA methods that do not require retraining: CoTTA achieves 25 fps on  $512 \times 512$  images (DAGM) and 9 fps on  $512 \times 1408$  images (KolektorSDD). This latency is due to the need to update both student and teacher models simultaneously (our method only requires updating one). For other methods that require retraining (unsupervised and weakly supervised), they can achieve faster inference speeds during the inference stage. However, the additional training takes approximately 15 min (KSDD) to 30 min (DAGM) and results in poorer inference accuracy (as shown in Table 4). By default, all of our results are based on a single Nvidia RTX2080Ti GPU.

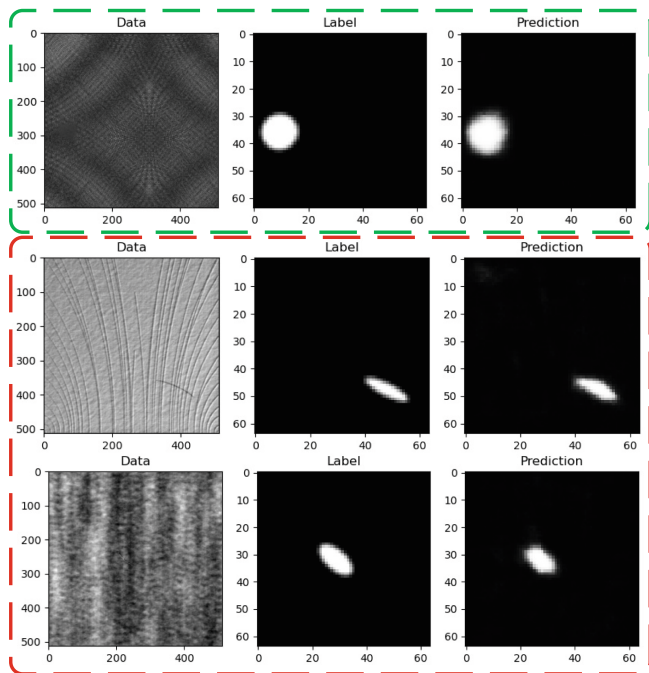
*Test Time Setting.* Without special emphasis, we set batch size=1 and use 1e-3 as the learning rate with Adam optimizer. Following [49], we use the same data augmentation operations, including color jitter, random affine, random horizontal flip, and so on. We use 4 augmentations for our experiments. The threshold  $p$  is 0.6 by default.

As shown in Table 3, The sample sizes of the surface-defect detection datasets are very small. This makes the base model originally supported by Cotta [49] and Tent [47] perform poorly on KolektorSDD. (On CIFAR10C [16] they used WideResNet-28 [57], on CIFAR100-C they used ResNeXt-29 [53] and on ImageNet-C [16] they used resnet50 [5]) For a fair comparison, we used the same two-stage model [2] as the base model, along with the same epoch training parameters to initialize. This allows us to more accurately compare the strengths and weaknesses of the methods in the adaptive phase. It is guaranteed that the difference is not due to a difference in the base model.

*Experiments on DAGM.* We first validate the effectiveness of our method on the DAGM dataset, which has ten classes, each with different texture and surface anomalies. To verify the reliability and stability of our method, we randomly select four of the ten classes as the source domain for training, while using the remaining six classes as the testing set for testing. This experiment was repeated for several sets. In Table 4, we present the full results comparing the accuracy of inference using our proposed method with inference directly on the testing set, thus demonstrating that our method can improve the inference accuracy of the model on the target domain when the source and target domains do not coincide. Also, as shown in Table 4, for a fair comparison, we compare our method with other TTA methods, demonstrating that our method is more applicable to industrial scenarios. We also compare with unsupervised and weakly supervised methods. For the unsupervised and weakly supervised work, we train on a training set for each class and test on a testing set. For unsupervised and weakly supervised methods, we follow [3]. For TENT [47] and CoTTA [49], we use the official open-source code. The results are averaged over all 10 classes. For the TTA work, we use a four-class training set for training and a ten-class test set for inference to demonstrate the model’s ability to remember source domain knowledge and adapt to the target domain. We demonstrate that our approach can make better use of the source domain knowledge, combined with

the unlabelled target domain knowledge, to obtain better inference accuracy on the target data domain. Figure 3 presents some examples of the predictions of our method.

In addition, we have found that TTA methods designed for traditional segmentation tasks do not achieve good accuracy in surface-defect detection. They are even significantly lower than the weakly supervised methods trained on the target domain from scratch. This is mainly due to their complex design (e.g. randomly recovering some of the model parameters as initialization) and their full update (updating the model parameters with every sample) that do not apply to this task. The high level of instability in the target domain, and the false pseudo-labeling produced, hurt the inference of the model.



**Fig. 3. Visualization of images, labels, and segmentations on the DAGM** The green box shows the effect of segmentation within the source domain. The red boxes show the segmentation of the new classes that emerged during the test-time (Color figure online)

*Experiments on KolektorSDD.* We validated the effectiveness of our method on the KolektorSDD dataset. Similar to the work done on DAGM, we compare our method with existing TTA methods [47], unsupervised [1, 40] and weakly supervised methods [3] as shown in the Table 5. The KolektorSDD dataset is simpler compared to the DAGM dataset, so we only performed a comparison

**Table 3.** Details of the evaluation datasets

Dataset	Positive Samples	Negative Samples	Defect Types	Annotations
DAGM1-6	450	3000	6	ellipse
DAGM7-10	600	4000	4	ellipse
KolektorSDD	52	347	1	bbox

**Table 4. Comparison with state-of-the-art SDD methods on the DAGM dataset.** AP, CA, US, and WS are abbreviations for Average Precision, Classification Accuracy, Unsupervised, and Weakly Supervised, respectively

Method	Venue	Type	CA	AP
f-AnoGAN [40]	MIA 2019	US	79.7	19.5
Uninf. stud. [1]	CVPR 2020	US	84.3	66.8
Staar [44]	CIRP 2019	US	–	–
CADN-W18 [58]	PR 2021	WS	86.2	–
CADN-W18(KD) [58]	PR 2021	WS	87.6	–
CADN-W32 [58]	PR 2021	WS	89.1	–
TNET [47]	ICLR 2021	TTA	86.3	85.1
CoTTA [49]	CVPR 2022	TTA	85.2	84.4
EATA [37]	ICML 2022	TTA	89.3	90.1
SAR [38]	ICLR 2023	TTA	87.9	86.1
DeYO [26]	ICLR 2024	TTA	90.4	90.6
Our method	TTA	–	90.3	89.2

of AP accuracy (most methods can achieve very high accuracy in classification accuracy). We visualize the prediction results in Fig. 4 to verify the effectiveness.

Although we manually filtered the dataset to partition the KolektorSDD dataset into widely varying source and target domains, this was still not sufficient to demonstrate the migration capability of our method. Therefore, we used four randomly selected categories in DAGM for training and then used this pre-trained model directly for adapting on the KolektorSDD dataset. A competitive accuracy of 0.93 was obtained for our model.

### 4.3 Ablation Study

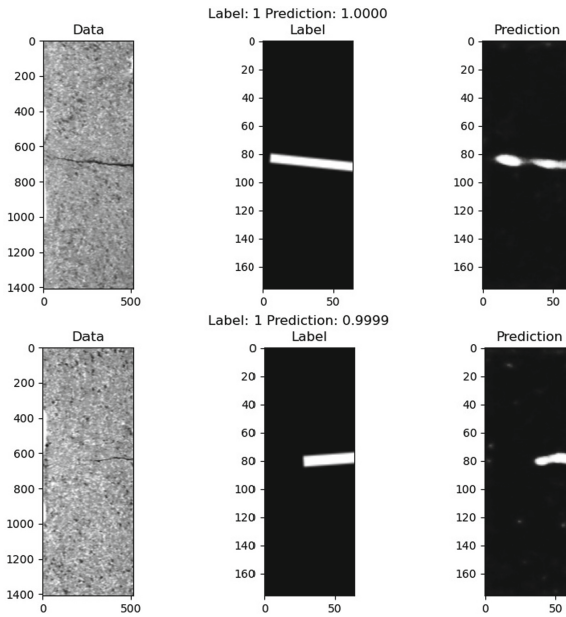
Finally, we evaluated the impact of each component, named supervisor filtering, augmented mean prediction, and dynamic balance loss. Results are reported in Table 6. We conducted ablation studies on the DAGM and KolektorSDD datasets. We used the same number of samples for testing, uniformly initialized with a pre-trained model trained for 50 epochs. We report performance by progressively enabling individual components and disabling specific components

**Table 5. Comparison with prior works on KolektorSDD dataset.** For unsupervised and weakly supervised methods, we follow the official codes of [3]. For tent [47]

Method	f-AnoGAN	Uninf. stud. [3]	TNET	Our method
AP	39.4	57.1	93.4	94.7

**Table 6.** Performance of individual components on DAGM dataset

AP	supervisor filtering	Augmented mean prediction	Dynamic balance loss
85.7			
87.2 ✓			
87.9 ✓		✓	
88.5 ✓		✓	✓



**Fig. 4.** Examples of predictions from the KolektorSDD

while retaining all remaining components. The results are reported in Table 4. The results show that on all three datasets, the worst performance was achieved with no components enabled, while the best performance was achieved with all three components. Below, we describe in detail the contribution of each component to the overall improvement.

The use of a supervisor to filter the sample data yields the greatest accuracy gain for this method. The large discrepancy between the source and target

domain data, along with the small sample data size and small model size, leads to a more pronounced accumulation of errors if incorrect pseudo-labels are used to train the model. Such errors can lead to even greater errors from subsequent pseudo-labels, resulting in worse model accuracy compared to direct inference in the target domain (without test-time adaptation). Using a supervisor to filter noisy samples mitigates this problem well.

All our designed components for test-time adaptation, including augmented mean prediction and dynamic balance loss, contributed to the accuracy improvement compared to testing directly on the target domain without adaptation. This demonstrates that the method we have designed is effective in improving the robustness of the model.

## 5 Conclusion

In this paper, we propose a novel online test-time adaptation framework for surface-defect detection, which addresses the challenge of detecting unforeseen anomalies in product surfaces during test time. We introduce the parameter-frozen supervisor to allow the model to remember the source domain knowledge over time, while continuously updating the model parameters to adapt to the distribution of the target domain. To bolster pipeline robustness, we devise augmented mean prediction and dynamically-balancing loss. Considering the difficulty and cost of collecting anomalous samples, our framework not only saves time and resources but also enhances the efficiency of the detection process. Thus, our method offers a promising solution to the surface-defect detection in industrial production processes. Experimental results demonstrate that our method yields superior inference accuracy on both the source and target domains.

**Acknowledgments.** Thanks to the support of Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200), National Natural Science Foundation of China (No. 72192821), YuCaiKe [2023] Project Number: 14105167-2023.

## References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: student-teacher anomaly detection with discriminative latent embeddings. In: CVPR, pp. 4183–4192 (2020)
2. Božič, J., Tabernik, D., Skočaj, D.: End-to-end training of a two-stage neural network for defect detection. In: ICPR, pp. 5619–5626. IEEE (2021)
3. Božič, J., Tabernik, D., Skočaj, D.: Mixed supervision for surface-defect detection: from weakly to fully supervised learning. *Comput. Ind.* **129**, 103459 (2021)
4. Chen, X., et al.: In: International Conference on Learning Representations, pp. 1–17 (2017)
5. Croce, F., et al.: Robustbench: a standardized adversarial robustness benchmark. In: NeurIPS Datasets and Benchmarks Track (2021)

6. Croitoru, I., Bogolin, S.V., Leordeanu, M.: Unsupervised learning from video to detect foreground objects in single images. In: International Conference on Computer Vision, pp. 4335–4343 (2017)
7. Duan, Y., Qi, L., Wang, L., Zhou, L., Shi, Y.: RDA: reciprocal distribution alignment for robust semi-supervised learning. In: ECCV, pp. 533–549. Springer (2022)
8. Duan, Y., et al.: Mutexmatch: semi-supervised learning with mutex-based consistency regularization. *TNNLS* **35**(6), 8441–8455 (2024)
9. Duan, Y., Zhao, Z., Qi, L., Zhou, L., Wang, L., Shi, Y.: Towards semi-supervised learning with non-random missing labels. In: ICCV, pp. 16121–16131 (2023)
10. Feng, Z., et al.: DMT: dynamic mutual training for semi-supervised learning. *PR* **130**, 108777 (2022)
11. Goodfellow, I., et al.: Generative adversarial nets. In: NeurIPS, pp. 2672–2680 (2014)
12. Gu, Q., et al.: Pit: position-invariant transform for cross-fov domain adaptation. In: ICCV, pp. 8761–8770 (2021)
13. Guo, S., et al.: Label-free regional consistency for image-to-image translation. In: ICME, pp. 1–6 (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
15. He, L., et al.: End-to-end video object detection with spatial-temporal transformers. In: ACM MM, pp. 1507–1516 (2021)
16. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019)
17. Hoyer, L., Dai, D., Van Gool, L.: Daformer: improving network architectures and training strategies for domain-adaptive semantic segmentation. arXiv preprint [arXiv:2111.14887](https://arxiv.org/abs/2111.14887) (2021)
18. Hu, M., et al.: Fully test-time adaptation for image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 251–260. Springer (2021)
19. Hu, X., et al.: Mixnorm: test-time adaptation through online normalization estimation. arXiv preprint [arXiv:2110.11478](https://arxiv.org/abs/2110.11478) (2021)
20. Iwasawa, Y., Matsuo, Y.: Test-time classifier adjustment module for model-agnostic domain generalization. In: NeurIPS, vol. 34, pp. 2427–2440 (2021)
21. Jiang, J., et al.: Dg-pic: domain generalized point-in-context learning for point cloud understanding. In: ECCV (2024)
22. Karani, N., Erdil, E., Chaitanya, K., Konukoglu, E.: Test-time adaptable neural networks for robust medical image segmentation. *MIA* **68**, 101907 (2021)
23. Kundu, J.N., Kulkarni, A., Singh, A., Jampani, V., Babu, R.V.: Generalize then adapt: source-free domain adaptive semantic segmentation. In: ICCV, pp. 7046–7056 (2021)
24. Kundu, J.N., Venkat, N., Babu, R.V.: Universal source-free domain adaptation. In: CVPR, pp. 4544–4553 (2020)
25. Kurmi, V.K., Subramanian, V.K., Namboodiri, V.P.: Domain impression: a source data free domain adaptation method. In: WACV, pp. 615–625 (2021)
26. Lee, J., et al.: Entropy is not enough for test-time adaptation: from the perspective of disentangled factors. In: ICLR (2024)
27. Li, R., Jiao, Q., Cao, W., Wong, H.S., Wu, S.: Model adaptation: unsupervised domain adaptation without source data. In: CVPR, pp. 9641–9650 (2020)
28. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. arXiv preprint [arXiv:1603.04779](https://arxiv.org/abs/1603.04779) (2016)



29. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: ICML, pp. 6028–6039 (2020)
30. Liang, K., et al.: A survey of knowledge graph reasoning on graph types: static, dynamic, and multi-modal. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 9456–9478 (2024)
31. Liu, F., et al.: CloudMix: dual mixup consistency for unpaired point cloud completion. *IEEE Trans. Vis. Comput. Graph.* (2024)
32. Liu, F., et al.: Emphasizing semantic consistency of salient posture for speech-driven gesture generation. In: ACM MM (2024)
33. Long, S., et al.: DGMamba: domain generalization via generalized state space model. In: ACM MM (2024)
34. Long, S., Zhou, Q., Ying, C., Ma, L., Luo, Y.: Diverse target and contribution scheduling for domain generalization. arXiv preprint [arXiv:2309.16460](https://arxiv.org/abs/2309.16460) (2023)
35. Long, S., Zhou, Q., Ying, C., Ma, L., Luo, Y.: Rethinking domain generalization: discriminability and generalizability. *IEEE Trans. Circuits Syst. Video Technol.* (2024)
36. Mummadi, C.K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., Metzen, J.H.: Test-time adaptation to distribution shift by confidence maximization and input transformation. arXiv preprint [arXiv:2106.14999](https://arxiv.org/abs/2106.14999) (2021)
37. Niu, S., et al.: Efficient test-time model adaptation without forgetting. In: ICML, pp. 16888–16905 (2022)
38. Niu, S., et al.: Towards stable test-time adaptation in dynamic wild world. In: ICLR (2023)
39. Prabhu, V., Khare, S., Kartik, D., Hoffman, J.: Sentry: selective entropy optimization via committee consistency for unsupervised domain adaptation. In: ICCV, pp. 8558–8567 (2021)
40. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks. *Med. Image Anal.* **54**, 30–44 (2019)
41. Song, Y., Zhou, Q., Li, X., Fan, D.P., Lu, X., Ma, L.: Ba-sam: scalable bias-mode attention mask for segment anything model. In: CVPR (2024)
42. Song, Y., Zhou, Q., Lu, X., Shao, Z., Ma, L.: Simada: a simple unified framework for adapting segment anything model in underperformed scenes. arXiv preprint [arXiv:2401.17803](https://arxiv.org/abs/2401.17803) (2024)
43. Song, Y., Zhou, Q., Ma, L.: Rethinking implicit neural representations for vision learners. In: ICASSP (2023)
44. Staar, B., Lütjen, M., Freitag, M.: Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP* **79**, 484–489 (2019)
45. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A.A., Hardt, M.: Test-time training for out-of-distribution generalization. [arXiv:1909.13231](https://arxiv.org/abs/1909.13231) (2019)
46. Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **31**(3), 759–776 (2020)
47. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: fully test-time adaptation by entropy minimization. In: ICLR (2021)
48. Wang, H., Liu, F., Zhou, Q., Yi, R., Tan, X., Ma, L.: Continuous piecewise-affine based motion model for image animation. In: AAAI, vol. 38, pp. 5427–5435 (2024)
49. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: CVPR, pp. 7201–7211 (2022)
50. Wang, X., et al.: TF-FAS: twofold-element fine-grained semantic guidance for generalizable face anti-spoofing. In: ECCV (2024)

51. Weimer, D., Scholz-Reiter, B., Shpitalni, M.: Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Ann. Manuf. Technol.* **65**(1), 417–420 (2016)
52. Wilson, G., Cook, D.J.: A survey of unsupervised deep domain adaptation. *ACM TIST* **11**(5), 1–46 (2020)
53. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: *CVPR*, pp. 1492–1500 (2017)
54. Xu, H., et al.: Semi-supervised 3D object detection via adaptive pseudo-labeling. In: *ICIP*, pp. 3183–3187 (2021)
55. Yang, H., Sun, H., Zhou, Q., Yi, R., Ma, L.: ZDL: zero-shot degradation factor learning for robust and efficient image enhancement. In: *CAD/Graphics*, pp. 266–280 (2023)
56. You, F., Li, J., Zhao, Z.: Test-time batch statistics calibration for covariate shift. arXiv preprint [arXiv:2110.04065](https://arxiv.org/abs/2110.04065) (2021)
57. Zagoruyko, S., Komodakis, N.: Wide residual networks. [arXiv:1605.07146](https://arxiv.org/abs/1605.07146) (2016)
58. Zhang, J., Su, H., Zou, W., Gong, X., Zhang, Z., Shen, F.: CADN: a weakly supervised learning-based category-aware object detection network for surface defect detection. *Pattern Recogn.* **109**, 107571 (2021)
59. Zhou, A., Levine, S.: Training on test data with bayesian adaptation for covariate shift. arXiv preprint [arXiv:2109.12746](https://arxiv.org/abs/2109.12746) (2021)
60. Zhou, F., Zhou, Q., Li, X., Lu, X., Ma, L., Ling, H.: Adversarial attacks on both face recognition and face anti-spoofing models. arXiv preprint [arXiv:2405.16940](https://arxiv.org/abs/2405.16940) (2024)
61. Zhou, F., Zhou, Q., Yin, B., Zheng, H., Lu, X., Ma, L., Ling, H.: Rethinking impersonation and dodging attacks on face recognition systems. In: *ACM MM* (2024)
62. Zhou, Q., et al.: Uncertainty-aware consistency regularization for cross-domain semantic segmentation. *CVIU* **221**, 103448 (2022)
63. Zhou, Q., et al.: Context-aware mixup for domain adaptive semantic segmentation. *TCSVT* **33**(2), 804–817 (2023)
64. Zhou, Q., Gu, Q., Pang, J., Lu, X., Ma, L.: Self-adversarial disentangling for specific domain adaptation. *TPAMI* **45**(7), 8954–8968 (2023)
65. Zhou, Q., et al.: TransVOD: end-to-end video object detection with spatial-temporal transformers. *TPAMI* **45**(6), 7853–7869 (2023)
66. Zhou, Q., Zhang, K.Y., Yao, T., Lu, X., Ding, S., Ma, L.: Test-time domain generalization for face anti-spoofing. In: *CVPR*, pp. 175–187 (2024)
67. Zhou, Q., et al.: Instance-aware domain generalization for face anti-spoofing. In: *CVPR*, pp. 20453–20463 (2023)
68. Zhou, Q., Zhang, K.Y., Yao, T., Yi, R., Ding, S., Ma, L.: Adaptive mixture of experts learning for generalizable face anti-spoofing. In: *ACM MM*, pp. 6009–6018 (2022)
69. Zhou, Q., et al.: Generative domain adaptation for face anti-spoofing. In: *ECCV*, pp. 335–356 (2022)
70. Zhou, Q., Zhuang, C., Yi, R., Lu, X., Ma, L.: Domain adaptive semantic segmentation via regional contrastive consistency regularization. In: *ICME*, pp. 01–06 (2022)



# Alleviating Catastrophic Forgetting in Facial Expression Recognition with Emotion-Centered Models

Israel A. Laurensi<sup>1</sup>(✉), Alceu de Souza Britto Jr.<sup>1</sup>, Jean Paul Barddal<sup>1</sup>,  
and Alessandro Lameiras Koerich<sup>2</sup>

<sup>1</sup> Graduate Program in Informatics (PPGIa), Pontifícia Universidade Católica do  
Paraná (PUCPR), Curitiba, PR, Brazil  
[israel.rosa@pucpr.edu.br](mailto:israel.rosa@pucpr.edu.br)

<sup>2</sup> École de Technologie Supérieure (ÉTS), Montréal, Canada

**Abstract.** Facial expression recognition is pivotal in machine learning, facilitating various applications. However, convolutional neural networks (CNNs) are often plagued by catastrophic forgetting, impeding their adaptability. The proposed method, emotion-centered generative replay (ECgr), tackles this challenge by integrating synthetic images from generative adversarial networks. Moreover, ECgr incorporates a quality assurance algorithm to ensure the fidelity of generated images. This dual approach enables CNNs to retain past knowledge while learning new tasks, enhancing their performance in emotion recognition. The experimental results on four diverse facial expression datasets demonstrate that incorporating images generated by our pseudo-rehearsal method enhances training on the targeted dataset and the source dataset while making the CNN retain previously learned knowledge.

**Keywords:** Facial expression recognition · Convolutional Neural Networks · Catastrophic forgetting · Pseudo-rehearsal · Regularization

## 1 Introduction

Emotions are essential in human interaction and comprehension. In such a context, facial expressions play an important role [15]. Thus, facial expression recognition (FER) is the functionality of numerous machine learning applications, including emotion-aware interfaces, personalized recommender systems, and human-robot interaction. One way to identify these emotions in complex systems is via convolutional neural networks (CNNs). These networks have achieved remarkable success in computer vision tasks such as image classification, object detection, and facial expression recognition. However, a significant limitation of CNNs is their susceptibility to catastrophic forgetting. When sequentially trained on different tasks or datasets, CNNs often struggle to retain previously learned information, which leads to degraded performance on previously mastered tasks.

This phenomenon impairs the practical application of CNNs in dynamic environments where models must continuously adapt to new data while retaining accuracy in the previous scenarios.

Evaluating the catastrophic forgetting problem in FER - a complex learning scenario - allows us to observe the proposed method's ability to deal with datasets composed of diverse emotional expressions, unlike more straightforward tasks with more limited patterns. Moreover, such an evaluation sheds light on the model's adeptness at maintaining previously learned emotional recognition performance while assimilating the changes of a new domain, showing faces collected with other acquisition protocols, and representing people with different characteristics and cultures.

Catastrophic forgetting arises due to CNN's optimization process, which tends to adjust the model's parameters to fit the current task, often overshadowing previously acquired representations. Researchers have proposed numerous approaches to mitigate catastrophic forgetting, including regularization techniques, dynamic neural network architecture, and rehearsal-based methods [7, 8, 12, 13, 16]. Furthermore, several literature reviews have been published in this research field and in continual learning, offering comprehensive insights into the state-of-the-art methodologies, best practices, and emerging trends in mitigating catastrophic forgetting and advancing continual learning algorithms [6, 11, 14]. While these state-of-the-art methodologies have demonstrated promising results in specific scenarios, they have limitations such as increased computational complexity or limited capacity to effectively retain information from past tasks, especially in facial expression recognition scenarios.

In this paper, we propose a novel approach to overcome the limitations of existing methods and effectively address catastrophic forgetting in CNNs when applied to facial expression recognition. Our approach capitalizes on generative adversarial networks (GANs) capabilities to generate synthetic samples that resemble the original training data. Incorporating these synthetic samples during training enables the CNN to re-learn and retain knowledge from previous tasks, thereby mitigating catastrophic forgetting. To achieve this, we generated synthetic images of each emotion (class) present in the datasets, aiming to better capture the intrinsic characteristics of each facial expression associated with human emotion. We refer to this method as emotion-centered generative replay (ECgr). Moreover, we introduce a quality assurance (QA) algorithm as a crucial component of our approach. The QA algorithm assesses the generated synthetic samples based on the CNN's original classification accuracy. Only high-quality synthetic samples, which the original CNN can accurately classify, are retained for training. This filtering step ensures that only superior generated samples are utilized, thus augmenting the performance of the proposed method. In addition, we weigh the importance of the synthetic images, considering the CNN output score as an image quality assignment. Such a weight penalizes images that have been assigned a low confidence value by the CNN, which might positively influence the training convergence, as these images may be considered detrimental to the adaptation to the new dataset.

Our hypothesis centers around the effectiveness of employing a pseudo-rehearsal method: H1) the utilization of a pseudo-rehearsal method, particularly our emotion-centered generative replay, offers a potential solution for memory decay in CNNs; H2) the fusion of our emotion-centered generative replay and the proposed QA algorithm offers a promising strategy to counteract memory decay within neural networks; and H3) the combination of emotion-centered generative replay, QA, and a weighted loss function is hypothesized to further strengthen memory retention and performance in neural networks, potentially surpassing the benefits of either ECgr or QA alone. To assess the proposed method’s efficiency and validate our hypothesis, we undertook facial expression recognition experiments across various emotion datasets and employed diverse training methodologies.

The contribution of this work is three-fold: i) a new pseudo-rehearsal method focused on the emotions to mitigate catastrophic forgetting when learning facial emotion recognition; ii) a loss function considering a penalization schema for low-quality synthetic images generated in the pseudo-rehearsal strategy; iii) a robust experimental protocol considering well-known FER datasets and a pipeline of experiments to discuss the contributions of the proposed emotion-centered generative replay in mitigating catastrophic forgetting when compared to a regular fine-tuning process or the possibility of joining datasets.

The remainder of this paper is structured as follows: Sect. 2 reviews related work on catastrophic forgetting and existing methods for its mitigation. Section 3 presents the proposed emotion-centered generative replay approach and outlines the architecture of the QA algorithm. Section 4 describes the experimental setup and presents the results of our comprehensive evaluations. Section 5 discusses the implications of our findings, and Sect. 6 concludes the paper, outlining potential directions for future research.

## 2 Related Works

Catastrophic forgetting has spurred numerous research works to minimize its effects. In this section, we explore prominent algorithms and insights inspired by neuropsychology, all aimed at addressing forgetting and improving memory retention within neural networks.

Learning without forgetting (LWF) stands out by employing knowledge distillation [8]. This technique transfers distilled knowledge from a model trained on prior tasks to a new model, thereby allowing the assimilation of new information while safeguarding the retention of past information. This intelligent utilization of previous knowledge effectively counteracts the plague of forgetting and amplifies the network’s overall performance. Another regularization method, elastic weight consolidation (EWC) [7], introduces a nuanced regularization term in the scenario. This term identifies and assigns significance to pivotal network parameters linked to previous tasks, penalizing alterations to these parameters during subsequent training phases. By preserving these key parameters diligently, EWC balances between accommodating novel tasks and upholding the wisdom derived from past experiences.

Synaptic intelligence (SI) [16] offers an innovative perspective that stems from evaluating past task performance and assigns weight to synaptic connections based on their influence. The more a synapse contributes, the higher its importance; in contrast, less influential synapses are assigned lower importance. By preserving these critical connections, SI bridges the gap between old and new information, thus mitigating forgetting while embracing novelty.

Deep generative replay (DGR) [12] utilizes generative models to create simulated instances from prior tasks during training. This approach effectively enriches the current task’s dataset. The augmented instances, fused with real-time data, offer the network a diverse and comprehensive pool of examples. With past knowledge seamlessly integrated, DGR effectively combats the erosion of previously gained insights, presenting itself as a powerful tool for memory retention.

Beyond these algorithms, insights obtained from neuropsychological research paint a broader picture. Investigations into context-dependent learning have illuminated the crucial role of training and testing contexts in determining network performance. Thus, using contextual cues, algorithms can be designed to exploit the training and testing context better, thereby enhancing memory retention while countering the forgetting phenomenon [11].

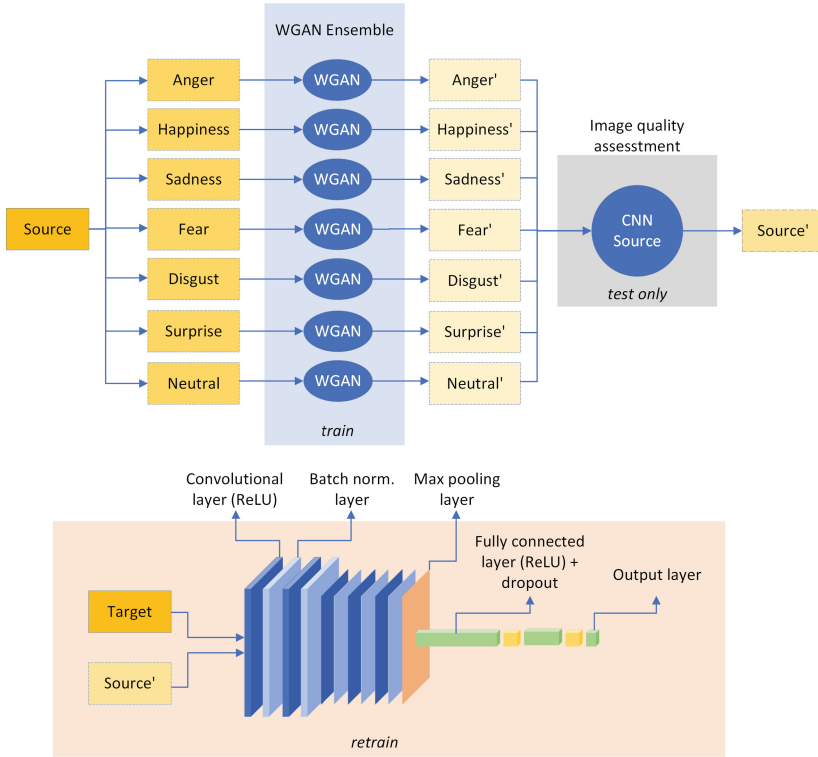
In light of these contributions, it is crucial to contextualize our work within the broader realm of the current state-of-the-art. The proposed methodology harmonizes the concepts of emotion-centered generative replay and QA. With CNNs as the focal point, our approach aims to prevent catastrophic forgetting in facial expression recognition, a domain where precise emotion identification heavily depends on image quality.

### 3 Proposed Method

In this section, we describe the methodology employed in our study to address the challenges of catastrophic forgetting in facial emotion recognition tasks. Our approach combines emotion-centered generative replay using a Wasserstein generative adversarial network with gradient penalty (WGAN-GP) and a QA algorithm. Figure 1 presents a general overview of the proposed method.

The use of WGAN-GPs is attributed to the stable learning power of these networks, a factor crucial when dealing with catastrophic forgetting. After all, attempting to address this issue through training and employing a generative method may lead to catastrophic forgetting in the generative networks. WGAN-GPs [5] implement a penalty on the gradient norm during training and optimization of the WGAN [2], thereby ensuring more stable training and yielding higher-quality generated images.

We have formalized our methodology using algorithmic representations to provide a more concrete understanding of the theoretical concepts presented. In Subsect. 3.2, we provide detailed algorithms replicating our approach’s offline preparation and training stages. These algorithms encapsulate the step-by-step processes of generating synthetic images and performing continuous retraining.



**Fig. 1.** An overview of the proposed method, separated into two key components. At the top, the emotion-centered WGAN-GP with CNN QA is depicted. This component involves training a WGAN-GP for each class in the source dataset to generate synthetic data resembling that class. At the bottom, the fine-tuning strategy is illustrated, where our synthetic dataset is replayed alongside the target dataset.

### 3.1 Emotion-Centered Generative Replay

We initiate by training a set of WGAN-GPs, one for each of the seven emotion classes present in the ‘source’ dataset - fear, anger, happiness, sadness, disgust, surprise, and neutral. Using these trained WGAN-GPs, we generate augmented datasets for each class. These generated images capture the intricate details of respective emotions, diversifying training data towards better generalization.

The WGAN-GP is built by two different networks: discriminator and generator. The discriminator network is crucial for distinguishing between real and synthetic images. It consists of several layers, including convolutional layers with leaky rectified linear unit (ReLU) activation functions. These layers help the discriminator extract relevant features from input images. Additionally, dropout layers are applied to prevent overfitting. This network contains approximately 4.3 million trainable parameters.

The generator network, detailed in Table 1, creates synthetic images from random noise. It uses dense, batch normalization, and convolutional layers with leaky ReLU activations to upscale and refine feature maps. The output matches the desired image size. With around 1.5 million parameters, this architecture produces images to challenge the discriminator.

**Table 1.** WGAN-GP generator and discriminator architecture.

Generator	Output Shape	Discriminator	Output Shape
Input Layer	(128)	Input Layer	(48, 48, 1)
Dense	(9216)	Zero Padding 2D	(52, 52, 1)
Batch Normalization	(9216)	Convolutional 2D	(26, 26, 64)
Leaky ReLU	(9216)	Leaky ReLU	(26, 26, 64)
Reshape	(6, 6, 256)	Convolutional 2D	(13, 13, 128)
Up Sampling 2D	(12, 12, 256)	Leaky ReLU	(13, 13, 128)
Convolutional 2D	(12, 12, 128)	Dropout	(13, 13, 128)
Batch Normalization	(12, 12, 128)	Convolutional 2D	(7, 7, 256)
Leaky ReLU	(12, 12, 128)	Leaky ReLU	(7, 7, 256)
Up Sampling 2D	(24, 24, 128)	Dropout	(7, 7, 256)
Convolutional 2D	(24, 24, 64)	Convolutional 2D	(4, 4, 512)
Batch Normalization	(24, 24, 64)	Leaky ReLU	(4, 4, 512)
Leaky ReLU	(24, 24, 64)	Flatten	(8192)
Up Sampling 2D	(48, 48, 64)	Dropout	(8192)
Convolutional 2D	(48, 48, 1)		
Batch Normalization	(48, 48, 1)		
Activation	(48, 48, 1)		
<b>Total params: 1,586,500</b>		<b>Total params: 4,303,360</b>	

We employ our QA algorithm to ensure the quality of the generated images. The QA algorithm filters out low-quality or incorrect images generated by the WGAN-GP, retaining high-quality images that the original classifier correctly classifies. The QA process is performed using the CNN trained on the source dataset. Given an empirically defined threshold, the images correctly classified by the network are used for future retraining, and the misclassified images are discarded. The QA process enhances the reliability of the emotion-centered generative replay, preventing the classifier from being influenced by poor-quality or misleading synthetic images. These images are then integrated into an improved dataset, which merges the synthetic images with the initial source data.

During retraining, the new dataset and the target dataset are employed. This unified dataset facilitates CNN training, where knowledge from the original emotion classes is combined with the new target emotions, minimizing forgetting.



### 3.2 General Pipeline

To address the challenge of catastrophic forgetting, our proposed approach involves a two-stage process: offline preparation and a training phase.

**Offline Preparation Stage.** Initialization occurs as depicted in Algorithm 1, where a set of datasets represented by  $T$  is defined, encompassing datasets  $A, B, C$ , and  $D$ . Each dataset  $d_t$  within  $T$  is traversed through an iterative process. For each specific dataset  $d_t$ , a classifier, denoted as  $C_{d_t}$ , is trained using that particular dataset.

---

#### Algorithm 1. Offline stage

---

```

1:  $T \leftarrow A, B, C, D$ 
2:  $T' \leftarrow \emptyset$ 
3: for each dataset  $d_t$  in  $T$  do
4:    $G_{d_t} \leftarrow \emptyset$ 
5:   Train classifier  $C_{d_t}$  on dataset  $d_t$ 
6:   for each class  $c$  in dataset  $d_t$  do
7:     Train  $\text{WGANGP}_c$  on class  $c$ 
8:     Add  $\text{WGANGP}_c$  to ensemble  $G_{d_t}$ 
9:     Generate  $\text{SI}_c$  using  $\text{WGANGP}_c$ 
10:    Pass  $\text{SI}_c$  through  $C_{d_t}$  to generate dataset  $d_{t_c}^{qa}$ 
11:    Add  $d_{t_c}^{qa}$  to dataset  $d'_t$ 
12:   end for
13:   Add  $d'_t$  to  $T'$ 
14: end for
15: return collection of synthetic datasets  $T'$ 

```

---

Our proposal then iterates over each class  $c$  in dataset  $d_t$ . In this context, a WGAN-GP is trained per class, denoted  $\text{WGANGP}_c$ , and these are subsequently combined to form an ensemble, denoted as  $G_{d_t}$ . Through these  $\text{WGANGP}_c$ , synthetic images ( $\text{SI}_c$ ) are generated to reflect the characteristics of each class. Continuing the process, these synthetic images are input to the classifier  $C_{d_t}$ , thus resulting in a new dataset,  $d_{t_c}^{qa}$ , consisting of the images that are correctly classified by  $C_{d_t}$ . These refined synthetic images are combined into a new dataset  $d'_t$ . This procedure is executed for each dataset  $d_t$ , and all the resulting datasets  $d'_t$  are unified into a collection labeled  $T'$ , encapsulating the sets of synthetic datasets corresponding to each original dataset  $d_t$  in  $T$ .

The time complexity of Algorithm 1 is  $O(n \cdot (f(p) + m \cdot g(p)))$ , where  $n$  is the number of datasets,  $m$  is the number of classes per dataset, and  $p$  is the number of images per class. The term  $f(p)$  represents the time complexity for training a classifier on  $p$  images, while  $g(p)$  denotes the complexity for training a WGAN-GP on  $p$  images.

**Continual Learning Stage.** Our approach began with individual training for ECgr before merging ECgr and QA. For a comparative evaluation, we utilize joint training and fine-tuning methods. Joint training simultaneously incorporates the source and target data while fine-tuning adapts the CNN to new data, training only the fully connected layers.

---

**Algorithm 2.** Continual learning stage

---

```

1:  $T \leftarrow B, C, D$ 
2:  $C_T \leftarrow \emptyset$ 
3: for each dataset  $d_t, d'_t$  in  $T, T'$  do
4:    $d_t^u \leftarrow d_t + d'_t$ 
5:   Train classifier  $C_{d_t^u}$  on unified dataset  $d_t^u$ 
6:   Add trained  $C_{d_t^u}$  to  $C_T$ 
7: end for
8: return ensemble  $C_T$ 

```

---

As shown in Algorithm 2, we define a set of subsequent datasets, indicated by  $T$ , which comprises datasets B, C, and D. Then, we iterate over each combination of the original dataset and subsequent dataset, referred to as  $d_t$  and  $d'_t$  respectively, from set  $T$  and its counterpart  $T'$ . For each dataset combination, we create a unified dataset,  $d_t^u$ , by merging  $d_t$  and  $d'_t$ . Subsequently, we train a classifier,  $C_{d_t^u}$  on the unified dataset  $d_t^u$ .

For Algorithm 2, the time complexity is  $O(n \cdot (r(m) + f(m)))$ . Here,  $n$  denotes the number of dataset pairs processed from sets  $T$  and  $T'$ , while  $m$  represents the size of individual datasets  $d_t$  and  $d'_t$ . The term  $r(m)$  stands for the overhead for merging datasets  $d_t$  and  $d'_t$ , whereas  $f(m)$  represents the computational cost of training a classifier on a dataset of size  $m$ .

The CNN used in our experiments, detailed in Table 2, begins with 2D convolutional layers (64 filters each) and batch normalization. It includes additional convolutional layers, max-pooling for downsampling, and further batch normalization for higher-level feature extraction. The feature maps are flattened and passed through fully connected layers with dropout to prevent overfitting. The final layer uses softmax activation to output class probabilities. Overall, this CNN architecture comprises approximately 19.3 million parameters.

The training aims to optimize the Eq. (1), where a weight  $w$  is applied to each prediction. This weight is determined by the CNN’s confidence percentage when predicting for all  $y_{pred}$ .

$$\mathcal{L}_i(\mathbf{y}_{\text{true}}^{(i)}, \mathbf{y}_{\text{pred}}^{(i)}) = - \sum_{j=1}^C w_j y_{\text{true } j}^{(i)} \log(y_{\text{pred } j}^{(i)}) \quad (1)$$

In summary, our general pipeline encompasses an offline preparation phase involving training WGAN-GPs and QA-based synthetic image generation. In the training stage, synthetic and original datasets are combined, and the continual

**Table 2.** CNN network architecture.

Layer (type)	Output Shape	Params
Convolution 2D	(47, 47, 64)	320
Batch Normalization	(47, 47, 64)	256
Convolution 2D	(46, 46, 64)	16448
Batch Normalization	(46, 46, 64)	256
Max Pooling 2D	(23, 23, 64)	0
Convolution 2D	(21, 21, 128)	73856
Batch Normalization	(21, 21, 128)	512
Convolution 2D	(19, 19, 128)	147584
Batch Normalization	(19, 19, 128)	512
Convolution 2D	(17, 17, 128)	147584
Batch Normalization	(17, 17, 128)	512
Max Pooling 2D	(8, 8, 128)	0
Flatten	(8192)	0
Dense	(2048)	16779264
Dropout	(2048)	0
Dense	(1024)	2098176
Dropout	(1024)	0
Dense (Softmax)	(7)	7175
<b>Total params: 19,272,455</b>		

retraining approach adapts the classifier to multiple datasets while incorporating different strategies.

## 4 Experiments

To evaluate the performance of our methodology, we utilize several datasets that contain human facial images displaying various emotions. The datasets considered in our study include TFEID, MUG, CK+, and JAFFE. These datasets provide diverse emotional contexts, allowing us to assess our approach’s robustness and generalization capabilities. All datasets have the following classes: fear, anger, happiness, sadness, disgust, surprise, and neutral.

The Multimodal Understanding Group (MUG) [1] dataset consists of approximately 1462 facial images, each annotated with the corresponding facial expression labels. The Japanese Female Facial Expression (JAFFE) [10] dataset, despite its relatively small size, containing approximately 213 facial images, is valuable for evaluating and comparing facial expression recognition models. The Taiwanese Facial Expression Image Database (TFEID) [3] provides a suitable testbed for evaluating emotion recognition algorithms, with 1128 samples.

Lastly, the extended Cohn-Kanade dataset (CK+) [9] is commonly used for facial expression recognition research. It includes a substantial number of facial images, compiled into 123 videos of different subjects, totaling approximately 593 videos, with 327 labeled videos covering various emotional expressions.

#### 4.1 Results

This section offers an in-depth analysis of the outcomes achieved by employing different retraining strategies, each suited to minimize memory degradation and maximize knowledge retention.



**Fig. 2.** Sample results for different classes from the MUG, JAFFE, and TFEID synthetic datasets generated by WGAN-GP. The first column (in green) displays the original samples from the MUG, TFEID, and JAFFE datasets (from top to bottom, respectively). In contrast, the second-to-last column (in orange) features the corresponding synthetic images for each dataset. (Color figure online)

**On Quality of Synthetic Images.** In this subsection, we present a comprehensive discussion of the qualitative aspects of the synthetic data. As shown in Fig. 2, the left side features an image from the original dataset as a reference for the dataset’s inherent visual characteristics. On the right side, seven columns display synthetic images generated for each class within the dataset. These columns show the diversity and fidelity of the synthetic samples produced by our ECgr approach. Figure 3 shows examples of images that were rejected during the QA process. These rejected images are of low quality and do not convey emotion, resulting in incorrect classification by the CNN.

#### 4.2 On Continual Learning.

In this section, we discuss the main results observed from the tests conducted with facial expression datasets, utilizing the combination of different methods outlined in this study.

Initially, we trained a CNN on the MUG dataset. We then adapted this CNN for continuous learning across other datasets. Each training process was replicated 20 times. For methods involving image generation, the synthetic datasets differ across various replications of CNN adaptation.



**Fig. 3.** Some rejected samples identified by the QA algorithm from the synthetic datasets of MUG, JAFFE, and TFEID.

In Tables 3, 4 and 5, the columns baseline, joint, and fine-tune represent, respectively: testing datasets with the CNN trained on the source dataset; adapting the CNN trained on the source plus target dataset; adapting the CNN trained on the source dataset using only the target dataset. Additionally, the ECgr and QA methods were evaluated separately (ECgr) and then combined (ECgr+QA) to determine the impact of using synthetic image filtering in continuous training. Furthermore, this scenario assessed whether using weights (ECgr+wQA) on synthetic images has any effect compared to training without this technique.

**Table 3.** Results on MUG’s model fine-tuned to JAFFE dataset in terms of ECgr, QA, weighted QA, and the combination of ECgr with QA and wQA, alongside with fine-tune, joint, and current for a direct comparison.

	Current model	Joining datasets	Fine Tuning	Proposed		
				ECgr	ECgr+QA	ECgr+wQA
Source dataset						
MUG	0.98 ± 0.00	1.00 ± 0.00	0.75 ± 0.03	0.88 ± 0.04	0.93 ± 0.02	0.94 ± 0.03
Target dataset						
JAFFE	0.28 ± 0.00	0.74 ± 0.06	0.77 ± 0.03	0.78 ± 0.03	0.78 ± 0.05	0.79 ± 0.04
Mean	0.63	0.87	0.76	0.83	0.85	0.86

Table 3 shows the results when adapting the CNN trained on the MUG dataset (source) to the JAFFE dataset (target). Considering the baseline, joint,

and fine-tune methods, we can assume that the upper limit is the joint method, which represents the ideal case where all datasets are available for training, and the lower limit is the fine-tune method, in which the source dataset is no longer available. The combined method ECgr+wQA yielded the best results in this initial adaptation involving only one dataset, with a result very close to joint training. Tables 4 and 5, show a change in this scenario, as more datasets are introduced in continuous training, the ECgr+QA method tends to outperform. Regarding the result obtained in the retraining for the JAFFE dataset, it is possible to justify this outcome, where the combined method (ECgr+QA) came close to the joint method, as the adaptation can still be considered trivial since only one dataset is being adapted. Thus, the complexity for the CNN to assimilate synthetic images needs to be higher.

**Table 4.** Results on MUG plus JAFFE’s model fine-tuned to TFEID dataset in terms of ECgr, QA, weighted QA and the combination of ECgr with QA and wQA, alongside with fine-tune, joint and current for a direct comparison.

	Current model	Joining datasets	Fine Tuning	Proposed		
				ECgr	ECgr+QA	ECgr+wQA
Source datasets						
MUG	$0.75 \pm 0.03$	$1.00 \pm 0.00$	$0.71 \pm 0.01$	$0.84 \pm 0.06$	$0.87 \pm 0.04$	$0.78 \pm 0.04$
JAFFE	$0.77 \pm 0.03$	$0.94 \pm 0.03$	$0.76 \pm 0.02$	$0.64 \pm 0.07$	$0.62 \pm 0.07$	$0.69 \pm 0.04$
Mean	0.75	0.97	0.73	0.74	0.74	0.73
Target dataset						
TFEID	$0.22 \pm 0.00$	$0.79 \pm 0.05$	$0.78 \pm 0.03$	$0.83 \pm 0.04$	$0.84 \pm 0.04$	$0.87 \pm 0.04$
Updated mean	0.58	0.91	0.75	0.77	0.78	0.78

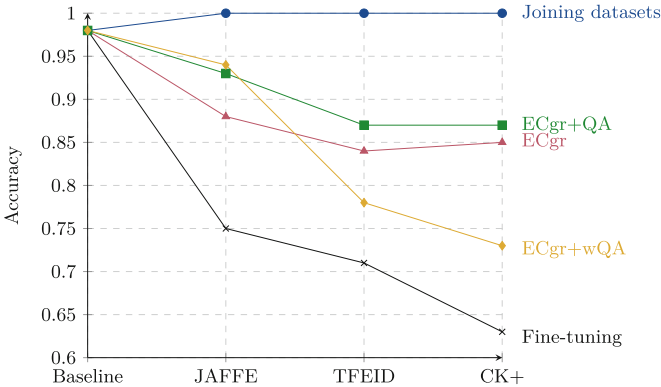
Table 4 shows, when adapting the CNN trained on MUG and JAFFE to the new dataset TFEID, that the best result lies between ECgr+QA and ECgr+wQA. Interestingly, in all results, the generative method - combined with QA or not - performed equally or better on the target dataset when compared to joint training. This reveals that synthetic images not only aid the CNN in recalling something it has already seen but also assist in training for new data, reinforcing knowledge when adapting to the same context, in this case, emotion recognition.

In Table 5, when adapting the CNN trained on MUG, JAFFE, and TFEID to CK+, we can observe a behavior similar to that observed when adapting to TFEID, where the best result lies between the ECgr+QA and ECgr+wQA methods. However, at this point, it becomes more apparent that using a weight for synthetic images brings an intrinsic problem to the training of the CNN being used for the filtering method. This CNN can carry certain behaviors into subsequent training steps, where errors from certain classes may compromise the entire training when using the confidence percentage.

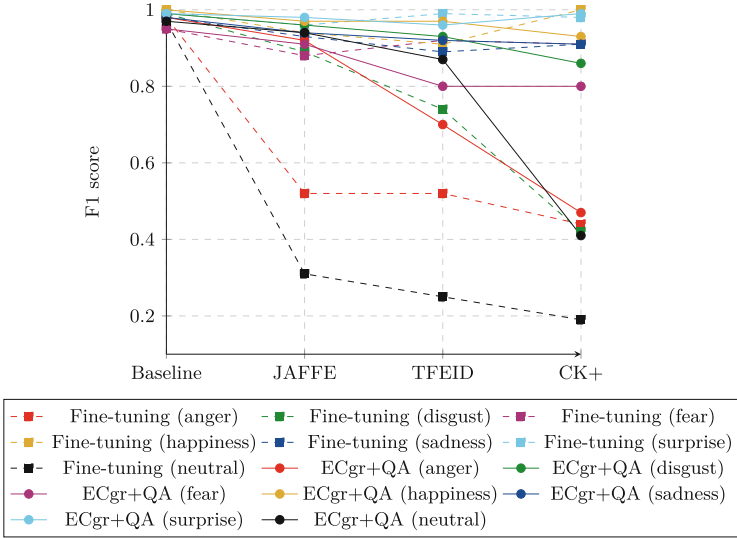
**Table 5.** Results on MUG plus JAFFE plus TFEID’s model fine-tuned to CK+ dataset in terms of ECgr, QA, weighted QA and the combination of ECgr with QA and wQA, alongside with fine-tune, joint and current for a direct comparison.

	Current model	Joining datasets	Fine Tuning	Proposed		
				ECgr	ECgr+QA	ECgr+wQA
Source datasets						
MUG	$0.71 \pm 0.01$	$1.00 \pm 0.00$	$0.63 \pm 0.05$	$0.85 \pm 0.04$	$0.87 \pm 0.03$	$0.73 \pm 0.04$
JAFFE	$0.76 \pm 0.02$	$0.99 \pm 0.01$	$0.57 \pm 0.08$	$0.61 \pm 0.05$	$0.55 \pm 0.04$	$0.59 \pm 0.05$
TFEID	$0.78 \pm 0.03$	$1.00 \pm 0.00$	$0.49 \pm 0.06$	$0.62 \pm 0.09$	$0.76 \pm 0.09$	$0.70 \pm 0.07$
Mean	0.73	0.95	0.56	0.69	0.72	0.67
Target dataset						
CK+	$0.53 \pm 0.00$	$0.81 \pm 0.03$	$0.79 \pm 0.03$	$0.83 \pm 0.03$	$0.82 \pm 0.03$	$0.81 \pm 0.02$
Updated mean	0.68	0.99	0.62	0.72	0.75	0.71

We can better understand the results in the MUG dataset from the continuous training of all datasets with Fig. 4. It is noticeable that the best method for the MUG dataset is ECgr+QA. We can also observe the poor performance of the fine-tuning method in the context of continuous training, where the knowledge was significantly forgotten compared to methods that attempt to mitigate this behavior. While fine-tuning initially shows promise in adapting the model to new tasks or domains, its performance deteriorates over time as knowledge retention becomes increasingly challenging. Additionally, memory forgetting becomes trivial when all datasets are always available, as datasets can be combined for retraining. However, one must consider the high computational cost and storage requirements of joint training.



**Fig. 4.** Accuracy results on the MUG dataset, showcasing the continuous adaptation of a trained CNN across JAFFE, TFEID and CK+ datasets relative to the baseline accuracy.



**Fig. 5.** Comparison of F1 scores by class on the MUG dataset between fine-tune and ECgr+QA, showcasing the continuous adaptation across JAFFE, TFEID, and CK+ datasets

Given that synthetic images for each class are generated independently in our method, it is essential to examine class-specific memory loss. Figure 5 compares the fine-tune and ECgr+QA methods, revealing subpar performance ( $F1 < 0.6$ ) for the anger and disgust classes. During the final retraining step, the ECgr+QA method also experiences performance deterioration for the fear class. This underscores the difficulty of training these classes, as even minor facial changes can be misinterpreted as another emotional state.

We have also conducted experiments on a different domain using the MNIST dataset, and the results are presented in Appendix A.

## 5 Discussion

Firstly, the results support our hypothesis regarding using pseudo-rehearsal methods, specifically emotion-centered generative replay, to minimize memory decay. Our strategy demonstrated remarkable efficacy in alleviating catastrophic forgetting, consistently outperforming the fine-tuning methods across various tasks. The generation of synthetic data resembling past task patterns through WGAN-GPs proved positive in enabling the network to retain knowledge without using original data. This substantiates our anticipation that pseudo-rehearsal techniques, particularly our emotion-centered generative replay, are essential in counteracting memory decay.

Furthermore, synthesizing our WGAN-GP class-driven generative and QA methods substantiates our second hypothesis. Introducing a QA mechanism dur-



ing replay significantly improved the quality of synthetic data, further augmenting the approach’s effectiveness. The third hypothesis, in which applying a weight to synthetic images would benefit continuous training, can only be observed as positive in the retraining for the first dataset - from MUG to JAFFE. We observed that this technique was ineffective for more datasets after JAFFE. This may be directly related to the errors of the network that assigns these weights to the synthetic images - meaning the network may be making errors with high confidence, negatively affecting the synthetic images, which in turn are not fully considered in the retraining, leading the CNN to not remember these data.

## 6 Conclusion

In this study, we presented a comprehensive investigation into the challenge of catastrophic forgetting in CNNs within the context of facial expression recognition, proposing a novel approach to mitigate its effects. We employed a pseudo-rehearsal method, specifically our emotion-centered generative replay (ECgr) with WGAN-GPs, to generate synthetic images for each dataset class and combined this with a filtering method to exclude images that could hinder retraining.

Across various tasks, ECgr consistently demonstrated superior performance compared to baseline and fine-tuned methods. Utilizing WGAN-GPs to synthesize task-specific data and our QA algorithm resulted in substantial knowledge retention. This confirms the potential of pseudo-rehearsal methods to effectively retrain CNNs without revisiting original datasets, offering a promising strategy for addressing memory decay, particularly in challenging scenarios like facial expression recognition.

Despite promising results with pseudo-rehearsal, its effectiveness may vary across network architectures, datasets, and tasks. Additionally, WGAN-GP-based data generation can be computationally expensive, limiting real-time use. These aspects highlight opportunities for future research, such as improved weight assignment algorithms and exploration of regularization techniques synergy with pseudo-rehearsal approaches. Also, enhancements can be made to the quality of images generated by the WGAN-GP and in the architecture of classifiers, for example, using transformer networks. While the primary concern remains mitigating catastrophic forgetting, there is significant potential to improve results by optimizing synthetic data usage.

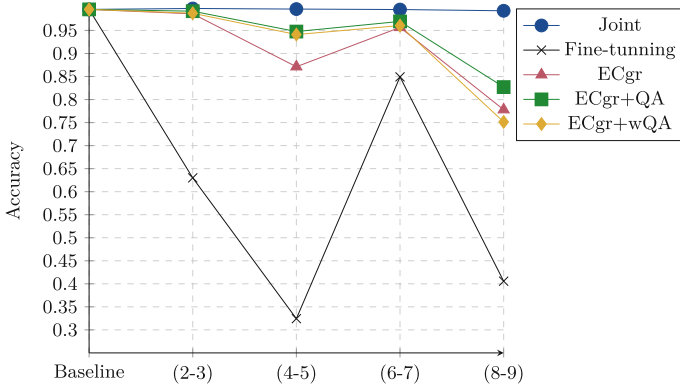
In an ideal scenario, combining classes from datasets such as MUG, JAFFE, and TFEID is recommended, as it enhances diversity and representation, leading to improved model performance. However, our method presents a viable alternative when such a combination is not feasible. This approach allows for flexibility in data augmentation and model training, providing a potential solution for scenarios with limited data availability or when data integration is challenging.

**Acknowledgments.** Thanks to CAPES SticAmSud (023-STIC-13), Univision Informática LTDA, Pontifícia Universidade Católica do Paraná (PUCPR) (grant 10844/2021) and CNPq (grant 306878/2022-4).

## A Appendix

### A.1 Evaluation of the MNIST Dataset

We evaluate our methodology across different domains using the MNIST dataset [4]. We applied the ECgr method with WGAN-GPs, dividing the dataset into class pairs (0 and 1, 2 and 3, 4 and 5, 6 and 7, 8 and 9), following the steps in Algorithm 1 and 2. Training began with the 0 and 1 class pair as the source dataset, with subsequent pairs used in the continual learning process. For continual learning, WGAN-GPs were trained for each digit, and the same process of combining target datasets with synthetic datasets generated by the generative networks was followed during retraining using the ECgr, ECgr+QA, and ECgr+wQA methods. As shown in Fig. 6, the behavior previously observed in FER datasets also held in this domain. The ECgr+QA and ECgr+wQA methods consistently outperformed fine-tuning in all retraining steps. Regarding the qualitative assessment of synthetic images, digits 4 and 5 were the most challenging to generate, and the QA algorithm struggled the most with these digits.



**Fig. 6.** Accuracy results for the MNIST (0-1) class pair subdataset, demonstrating continuous adaptation across subdatasets (2-3), (4-5), (6-7) and (8-9) relative to the baseline accuracy.

**Time Complexity.** Time and computational complexity were evaluated on an Intel Core i7-8700 CPU and an NVIDIA GeForce GTX 1060 GPU. The algorithm took approximately 5200 s to complete 20 replications of a single CNN retraining on the MNIST dataset. Each batch, with 1024 images, took 3 to 5 s to process. Predictions for 1000 images took approximately 3 s.

## References

1. Aifanti, N., Papachristou, C., Delopoulos, A.: The mug facial expression database. In: 11th International Workshop on Image Analysis for Multimedia Interactive Services, pp. 1-4 (2010)

2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Precup, D., Teh, Y.W. (eds.) 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 214–223. PMLR (2017)
3. Chen, C.C., et al.: A facial expression image database and norm for Asian population: a preliminary report. In: Farnand, S.P., Gaykema, F. (eds.) Image Quality and System Performance VI, vol. 7242, p. 72421D. SPIE (2009)
4. Deng, L.: The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**(6), 141–142 (2012). <https://doi.org/10.1109/MSP.2012.2211477>
5. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
6. Khetarpal, K., Riemer, M., Rish, I., Precup, D.: Towards continual reinforcement learning: a review and perspectives. *J. Artif. Intell. Res.* **73**, 295–333 (2022)
7. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci.* **114**(13), 3521–3526 (2017)
8. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(12), 2935–2947 (2018)
9. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: *IEEE CVPR Workshops*, pp. 94–101 (2010)
10. Lyons, M., Kamachi, M., Gyoba, J.: The Japanese female facial expression (JAFFE) dataset. Zenodo (1998). <https://doi.org/10.5281/zenodo.3451524>
11. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: a review. *Neural Netw.* **113**, 54–71 (2019)
12. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Guyon, I., et al. (eds.) NIPS, vol. 30. Curran Associates, Inc. (2017)
13. Tannugi, D.C., Britto, A.S., Koerich, A.L.: Memory integrity of CNNs for cross-dataset facial expression recognition. In: *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3826–3831 (2019)
14. van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three types of incremental learning. *Nat. Mach. Intell.* **4**(12), 1185–1197 (2022)
15. Zavaschi, T.H., Britto, A.S., Jr., Oliveira, L.E., Koerich, A.L.: Fusion of feature sets and classifiers for facial expression recognition. *Exp. Syst. Appl.* **40**(2), 646–655 (2013)
16. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. *Proc. Mach. Learn. Res.* **70**, 3987–3995 (2017)



# Satellite State Prediction and Maneuver Detection Analysis Using NCDEs

Kangjun Lee  and Simon S. Woo <sup>(✉)</sup> 

Sungkyunkwan University, Suwon, South Korea  
{gkd1677, swoo}@g.skku.edu

**Abstract.** Satellite Orbit Propagator (SOP) is of prime importance in the prevention of collision and completion of the assigned task of the satellites. In the past, orbit prediction and propagation have relied on physics-based mathematical models. However, as the number of satellites and their data increases, it is crucial to explore the data-driven orbit propagation based on advanced machine learning methods. In this work, we propose a novel deep learning-based framework to forecast future satellite orbit states. The proposed framework employs a model based on Neural Controlled Differential Equations (NCDEs) to train orbit prediction models, and our approach captures features from past satellite state values at both fixed and dynamic time intervals. The experimental results on Korea Aerospace Research Institute (KARI)'s KOMPSAT-3 and 5 datasets demonstrate that the proposed framework outperforms the other eight data-driven baseline forecasting models.

**Keywords:** Satellite Orbit Propagation · Satellite Orbit Maneuver Detection · Orbit Forecasting · Time Series Analysis

## 1 Introduction

Satellites adjust their orbits in response to changes in the space environment, such as air density change and collisions with space debris. The probability of collision has increased with the increasing number of satellites in the space environment, resulting in orbital changes that can cause unexpected changes in satellite paths. Therefore, it is of utmost importance to conduct satellite orbital analysis to prevent such collisions with different Resident Space Objects (RSOs). To address the aforementioned challenges, we employ the following two approaches: 1) Satellite Orbit Propagator (SOP), which focuses on forecasting satellite trajectories using orbital elements, and 2) Satellite Orbit Maneuver Detector (SOMD), which includes detecting satellite altitude maneuvers as well as known RSOs [23]. Those are key components for successful operation of satellites. While physics-based models approaches have been the most frequently utilized tools for predicting satellite orbits in the past, they sometimes produce high propagation

errors due to limitations in accurately modeling the unknown environments, and unexpected environments. In order to build an accurate physics-based model, understanding the target RSO’s space environment and body characteristics is required. Unfortunately, datasets and measurements in the space environments are often sparse and noisy, which directly affect the performance of a physics-based model, failing to achieve the required performance and accuracy. On the other hand, ML and data-driven approaches [20] have explored historical data patterns and correlations between numerous orbital components. Recently, data-driven and DL approaches for forecasting time series and anomaly detection have demonstrated satisfactory performance [7, 14, 24–26].

To effectively manage and predict the satellites’ orbital propagation and maneuvers, we propose the Satellite State Prediction and Maneuver Detection Analysis Framework (SSPMDA) based on Neural Controlled Differential Equations (NCDEs), an end-to-end framework for forecasting the satellite’s orbital elements and detecting maneuvers including visualization and analysis to aid the improved explainability. In particular, the Korea Aerospace Research Institute (KARI) is a national research center for researching and developing aerospace technologies, currently operating multipurpose and geostationary meteorological satellites. Our goal is to improve the robustness and overcome the limitations of physics-based model orbit propagation and maneuver detection missions. SSPMDA is composed of two main components: 1) the SOP, which forecasts the satellite’s future state of motion, and 2) the SOMD, which detects both known and unknown orbital maneuvers. We construct the forecasting model using the NCDEs structure and apply the trained weights to the SOMD to detect the maneuvers. Additionally, we use Spectral Residual from Ren et al. [18] to define the Maneuver Detection Score (MDS) to better focus on the satellite’s rapidly changing altitudes when identifying the maneuvers. For evaluation, we applied our SSPMDA to predict orbital elements on two different satellites of KOMPSAT-3 (K3) and 5 (K5) [12] from Jan. 2018 to Dec. 2019. The main contributions of our work are summarized as follows:

- We propose SSPMDA, a novel deep learning and NCDE-based forecasting-based framework for satellite orbit propagation and maneuver detection, where our method incorporates skip connections in NCDEs to enhance information propagation and increase the model’s expressiveness.
- The SSPMDA can forecast dynamic timespans using SOP. Furthermore, we detect satellite maneuvers with the SOMD by reusing the pre-trained weights of SOP and calculating the MDS, which focuses on forecasting errors of SOP.
- Extensive experiments on the real-world KARI satellite datasets demonstrate that our method achieves a prediction error of less than 0.59 km and outperforms other baselines.

## 2 Related Work

First, Melvin. [15] utilized the Kalman Filter (KF) in the field of stellar navigation to determine the orbit of satellites. The KF utilizes the gravity gradient

tensor to forecast the satellite’s orbit, incorporates it into the state transition matrix, and further employs them to propagate error covariances. However, real-world satellite orbit datasets are subject to noise, which adversely affects the performance of the KF. This leads to significant propagation errors and inaccurate orbit predictions. Moreover, Peng et al. [17] utilized Support Vector Machine (SVM) for satellite orbit prediction. To achieve accurate orbit prediction, they used publicly available data, Two-Line Element (TLE) catalog, and International Laser Ranging Service (ILRS) catalog [16]. However, the SVM used in this study is not suitable for predicting satellite orbits as it is difficult to capture the time dependency of multivariate time series. Several deep learning-based approaches have been proposed. USAD [1] is a method that uses an autoencoder architecture with one encoder and two decoders. It learns through adversarial training to perform anomaly detection based on reconstruction. TranAD [22] is a Transformer-based model that uses self-conditioning to extract multi-modal features and supports adversarial training for enhanced generalization. This model grows context information and supports temporal attention by utilizing position encoding. We use these anomaly detection models as a baseline to compare their effectiveness in maneuver detection. These existing methods have the shortcomings of being sensitive to noise, failing to capture temporal dependencies in multivariate time series, and focusing on anomaly detection rather than the specific challenges of satellite orbit prediction.

Also, there is an increasing interest in integrating differential equations and neural networks into models [19] for time series analysis. Among them, NCDEs are an extension of Neural Ordinary Differential Equations (NODEs) [3], combining the advantages of differential equations and neural networks. NODEs parameterize hidden vectors to predict time series data by modeling the continuous dynamics. Methods such as Euler’s method and Runge-Kutta method have been proposed to solve ordinary differential equations (ODEs) [2]. Unlike NODEs, which model continuous time series data for any time interval as a differential equation, NCDEs extend this capability to partially observed irregularly-sampled multivariate time series. To achieve this, it models and utilizes the Riemann-Stieltjes integral. While NODEs can be seen as a continuous analog of ResNet by formulating it as an ODE, NCDEs serve as a continuous analog of RNN. Moreover, NCDEs offer advantages in estimating unobserved data points by interpolating between discrete observations. Also, they can be integrated with various neural network architectures due to the model’s flexibility. We used GRU-ODE [5] as a baseline model to evaluate the performance of NCDEs.

### 3 Proposed Method

#### 3.1 Dataset

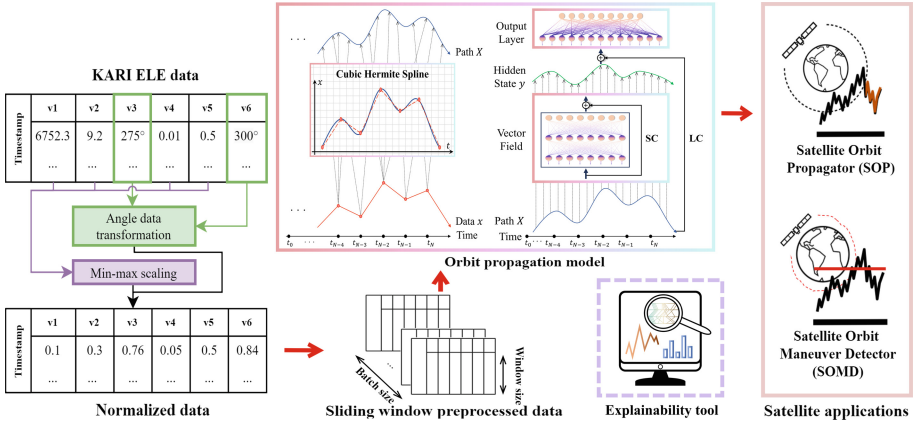
Our approach utilizes the precise orbits of the satellite datasets consisting of orbital elements collected from the KARI. In fact, KARI has been operating K3

and K5 [12] low earth orbit satellites since their launch in 2012 and 2013, respectively. In this work, we use precise orbital elements of K3 and K5 multipurpose satellites collected from Jan. 2018 to Dec. 2019. We refer to these datasets as the KARI satellite precision orbit Elements (ELE) datasets. The ELE consist of past satellite orbit data with six elements: semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, and mean anomaly. Since the ELE data for K3 and K5 share the same features, it is possible to design a single generalized model that performs well in predicting for all satellites. For ELE data, periodicity occurs, making altitude changes and maneuvers detection difficult, which in turn makes orbit propagation challenging with physics-based mathematical models. We explain that the ELE orbital elements consist of the following six variables in more detail:

1. **Semi-major axis** ( $a$ ) is the size of the orbit in kilometers, which is half of the longest diameter of the elliptical orbit. In practice, one can yield this size through the average of the periapsis and apoapsis distances.
2. **Eccentricity** ( $e$ ) refers to the degree of deviation from a circular shape for the orbit's shape and size, represented by a real number between 0 and 1, indicating the elliptical shape of the orbit.
3. **Inclination** ( $i$ ) refers to the angle between the equatorial plane and the orbital plane.
4. **Right ascension of the ascending node** ( $\Omega$ ) is an orbital parameter that describes a satellite's position in orbit relative to the Earth. It measures the angle between the vernal equinox and the point where the satellite's orbit crosses the equatorial plane while moving from south to north.
5. **Argument of perigee** ( $\omega$ ) is the angle measured from the ascending node to the perigee, defining the orientation of the ellipse in the orbital plane and indicating the point where the satellite reaches its closest distance to the Earth.
6. **Mean anomaly** ( $v$ ) is an orbital parameter used to describe the location of a satellite in its orbit around a central body. It represents the angular distance between the satellite and the perigee, indicating its position in orbit assuming a perfectly circular orbit.

### 3.2 Data Preprocessing Step

We employ two types of preprocessing methods: 1) min-max normalization and 2) the scaled sine transformation, according to the characteristics of the elements, as they have a unique range of values. First, we apply min-max normalization for each continuous element  $ele_c \in \{a, e, i, \omega\}$  as follows:  $f_{MinMax}(S_{ele_c}) = \frac{S_{ele_c} - \min(S_{ele_c})}{\max(S_{ele_c}) - \min(S_{ele_c})}$ , where the  $S_{ele_c}$  represents the series of the  $ele_c$  in the ELE datasets. Secondly, the right ascension of the ascending node and mean anomaly are expressed as angles between  $0^\circ$  and  $360^\circ$ . Their value increases to  $360^\circ$  and then becomes  $0^\circ$ , resulting in discontinuous data that can degrade the performance of machine learning and deep learning training procedures. Therefore, we apply the scaled sine transform to discontinuous elements  $ele_d \in \{\Omega, v\}$  as follows:  $f_{sin}(S_{ele_d}) = \frac{\sin(S_{ele_d}) + 1}{2}$ .



**Fig. 1.** The overall structure of our framework, SSPMDA. Our framework consists of interpolation and preprocessing of the orbit element data, forecast models, two different satellite applications (i.e., Satellite Orbit Propagator (SOP) and Satellite Orbit Maneuver Detection (SOMD)), and an explainability tool for data analysis. SC denotes short skip connection, and LC denotes long skip connection.

### 3.3 SSPMDA Architecture

Our analysis framework forecasts  $L_f$  future orbital elements from observing  $L_p$  historical elements and identifies satellite altitude maneuvers by reusing the weights of the forecasting model. Our method comprises several components, which are described in Fig. 1. In our framework, the KARI ELE data passes through a preprocessing phase described in Sect. 3.2. The preprocessed data is segmented into data of size  $L_s$  using a sliding window to form a dataset. This dataset is processed through the orbit propagation model, modeled by NCDEs, to perform SOP and SOMD tasks. Furthermore, to be deployed in real-world settings, we develop the explainability and interpretation tool, which is explained in Sect. 5.

**Satellite Orbit Propagator (SOP).** At a specific time  $t$ , SOP forecasts ELE data of  $[t + 1, t + 1 + L_f]$  with ELE data of  $[t - L_p, t]$ . SOP proceeds with multivariate time series forecasting that receives the input of shape  $(b, L_p, n)$  and generates the output of shape  $(b, L_f, n)$ , where  $b$  is the batch size, and  $n$  is the number of orbital elements (we set  $n$  as 6). The objective of the training process is to minimize the mean squared error (MSE) loss, which compares the prediction of the SOP with the ground truth (future element data) as follows:  $E_{mse} = \frac{1}{L_f} \sum_{i=1}^{L_f} (y_i - \hat{y}_i)^2$ , where  $y_i$  represents the future ELE data at time  $i$  and  $\hat{y}_i$  represents the forecasted orbital elements.

**Satellite Orbit Maneuver Detector (SOMD).** The purpose of SOMD is to preemptively detect maneuvers in orbit to avoid collisions with other



space objects, and to prevent satellites from abruptly changing their orbits. After obtaining the prediction results from the SOP forecasting model, we use the model’s trained weights to identify orbit maneuvers. In real-world, data with orbit maneuvers are more sparse than collected during regular operations without maneuvers. Therefore, we concentrate on developing the model in an unsupervised manner, utilizing the model weights learned with a normal operating duration (the period with fewer maneuvers). The main idea of SOMD is to use the forecasting error  $(y - \hat{y})^2$  for all orbital element samples. As the SOP model is trained with normal operational data, the model will generate a higher error when the input samples include maneuvers compared to those without maneuvers.

**Maneuver Detection Score with Spectral Residuals.** We define the MDS to differentiate the maneuvers from normal operations and predict the orbit maneuvers accurately. The orbit maneuver is closely related to the altitude, which we can calculate using the semi-major axis (distance from the center of the Earth). To obtain the MDS at time  $t$ , we first calculate the semi-major axis prediction error  $E_{semi}^t$  for an input sample of the semi-major axis at time  $t$  by comparing the forecasted semi-major axis and the ground truth. We further apply Spectral Residual (SR) [18] to the  $E_{semi}^t$  to highlight the sudden shift in the MSE, which indicates a significant change in the semi-major axis. In fact, Ren et al. [18] introduced saliency detection in the time series domain, utilizing residuals in the frequency domain to focus on the most significant part. The SR algorithm begins by converting the MSE to the frequency domain using Fourier Transform (FT) and then computing the SR. Finally, we use the Inverse FT to derive the time series representation of the SR map. The residual is computed by subtracting the average spectrum from the given semi-major axis input sequence’s frequency domain, defined as follows:

$$E_{semi}^t = (SOP(S_{semi}^{[t-L_p, t]}) - S_{semi}^{[t+1, t+1+L_f]})^2, \quad (1)$$

$$MDS^t = Spec(E_{semi}^{[t, t-L_s]}), \quad (2)$$

where  $Spec$  indicates the SR function, SOP is the trained Satellite Orbit Propagator model,  $S_{semi}$  is the given semi-major axis sequence, and  $L_s$  is the sliding window size of the SR function. If MDS exceeds a predefined threshold, SOMD will detect the satellites’ orbital movement at time  $t$  as an altitude maneuver.

**NCDEs Modeling.** NCDEs [13] are designed by combining neural networks and differential equations to model and predict time series data. In contrast to RNNs commonly used for time series prediction, NCDEs estimate continuous dynamics from hidden vectors as time progresses. This model fundamentally utilizes the concept of the Riemann-Stieltjes integral to handle time series data that is irregularly sampled or partially observed. Moreover, a cubic spline is

employed to interpolate the discrete values of a time series into a continuous time series path. The equation for NCDEs used in our work is provided below, where  $X(t)$  represents the cubic spline path:

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t); \theta_f) dX(t), \quad (3)$$

$$= z(t_0) + \int_{t_0}^{t_1} f(z(t); \theta_f) \frac{dX(t)}{dt} dt, \quad (4)$$

where the  $z(t)$  denotes the hidden vector of a multivariate time series, and the initial value problem is addressed using Eq. 3, with the initial vector  $z(0)$  [11]. In our work, Cubic Hermite splines are specifically used for NCDEs, where they exhibit flexibility and robustness by combining neural networks and differential equations for modeling our orbit propagation data. In particular, NCDEs have shown to achieve outstanding performance in multivariate time series prediction due to their capability to capture the dynamic patterns of complex time series data. Therefore, we apply NCDEs in SSPMDA. Moreover, skip connections are utilized to enhance the performance of the SSPMDA for SOMD [4,27]. To achieve this, both long skip connections (LC) and short skip connections (SC) are utilized, where the LC is applied to the internal forward part of the NCDEs to perform concatenation operations. On the other hand, the SC performs element-wise addition within the vector field during the forward pass in our SSPMDA.

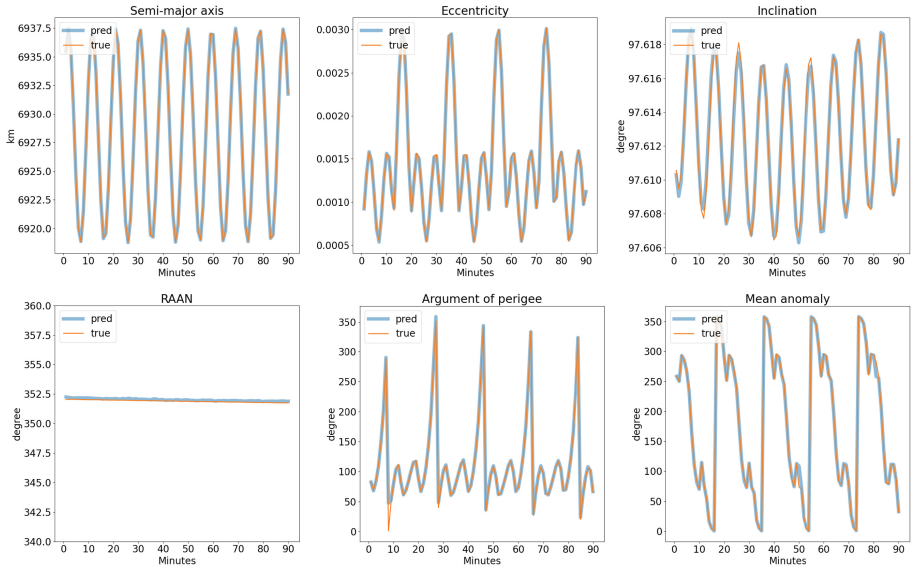
In SSPMDA, interpolation is performed prior to learning the data. This technique is used to estimate values for irregularly sampled or missing data points in time series datasets. In this work, we utilize the Cubic Hermite spline to fill discrete time series data and create a continuous path. We use cubic polynomials to transform each feature of the Normalized ELE data into a smooth continuous curve that passes through the data points. The Dormand-Prince method (DOPRI method) is utilized as a solver to numerically approximate the solution of NCDEs for the continuous path. The resulting initial condition and set of parameters pass through a vector field composed of a feed-forward neural network with ReLU, and the hidden state emerges as an output through the output layer. This output time series enables nearly precise prediction of satellite orbits and preemptive detection of maneuvers. In particular, our framework employs the Cubic Hermite spline to divide discrete data of K3 and K5 into smaller intervals and fit each interval with a cubic polynomial. The coefficients are determined by solving equations imposed by the given conditions, thereby obtaining data points. In this case, the Cubic Hermite spline corresponds to  $X(t)$  in Eq. 3. In our work, the use of Cubic Hermite splines allows for the integration of differential information, enabling precise control over the shape of the curve in K3 and K5 datasets. This makes it particularly useful for accurately capturing rapid changes in the data surpassing the capabilities of traditional cubic splines in orbit prediction problems.

## 4 Empirical Evaluations

**ELE Dataset.** We used six precise orbital elements of K3 and K5 multipurpose satellites, including 210,240 samples at a five-minute interval from Jan. 2018 to Dec. 2019. We split the ELE K3 and K5 datasets by year, using data from 2018 (105,120 samples) for training and data from 2019 (105,120 samples) for testing. Note that we evaluated the SOMD using only the K5 dataset (34 maneuvers in 2019), as the K3 satellite did not perform any maneuvers in 2019.

**Baselines.** We compared various data-driven baseline models against our SSPMDA model to demonstrate orbit propagation and maneuver detection performance: Linear Regression (LR), XGBoost, LSTM, Deep Belief Network (DBN), Seq2Seq with GRU cells, USAD, TranAD, and GRU-ODE. We used a multi-output regressor for LR. For XGBoost, we used the official implementation with 100 estimators. In our experiments, LSTM and Seq2Seq models were configured with four layers and 256 hidden states. Specifically, Seq2Seq utilized GRU cells. DBN consisted of a single hidden layer with 100 hidden units. USAD used the hidden units of each layer to 100. For TranAD, we implemented it according to the original code. GRU-ODE employed 256 hidden units.

**Implementation.** The duration of communication with the K3 and K5 satellites above South Korea is extremely short due to the satellites' high orbital speed (around 90 min orbital cycle), which makes developing reliable and accurate satellite operations challenging [9]. To address this challenge, we set the shortest input ( $L_p$ ) and prediction sequence length ( $L_f$ ) to 20 minutes, with the performance maintained when developing the prediction model. To model SSPMDA in an unsupervised manner, we utilize ELE data composed solely of normal data during training. As explained in the previous Sect. 3.3, if  $[t-L_p, t]$  is received as input, it is designed to predict  $[t+1, t+1+L_f]$ . In our proposed SSPMDA, the setup employs the Cubic Hermite interpolation as the data cubic spline method and designates the dopri5 [6] as the solver. Within the vector field of our model's architecture, we utilize two hidden linear layers with sizes of 32 and 64 units, respectively. We employ a *ReLU* activation function between each hidden layer for non-linearity and apply a *tanh* activation function at the final layer for output normalization. In our experiments, we trained the baseline models and our SSPMDA from scratch, minimizing the MSE objective function. XGBoost was trained with a GPU after specifying a maximum of 7 tree depths, a minimum of 5 instance weights, and a learning rate of 0.05. Deep learning (DL) models such as LSTM, Seq2Seq, USAD, TranAD, GRU-ODE, and SSPMDA are optimized using the Adam optimizer with a learning rate of 0.001. We trained DL models, including SSPMDA, for a maximum of 100 epochs and selected the best epoch based on the best validation MSE. In the case of our proposed framework, SSPMDA, skip connection is applied to improve SOMD, and this is covered in detail in the ablation study.



**Fig. 2.** Visualization of the prediction and ground truth derived from the six orbital elements of a satellite orbit cycle, where the X-axis indicates minutes, the Y-axis is the kilometer for the semi-major axis, the degree for the inclination, the ascension of the ascending node (RAAN), the argument of perigee, and mean anomaly.

#### 4.1 Experimental Results

The orbit prediction results of baselines and SSPMDA on K3 and K5 ELE datasets are reported in Table 1, where the best performance values are highlighted in bold. The “Total” column reports the average MSE of all orbital elements. In Table 1, we observed that SSPMDA achieves the lowest MSE (lower the better) at K3 and K5. In order to compare the ground truth and the predicted results intuitively, we rescaled the normalized results with a min-max scaling and the scaled sine transform in Sect. 3.2. The rescaled prediction results for K5 are visualized in Fig. 2. As shown in Fig. 2, the error between the predicted results and the ground truth is negligible. Especially the semi-major axis of all models has an error within 0.59 km and 0.09 km on K3 and K5, respectively.

We also compared all baseline models with our SSPMDA on detecting the maneuvers of the K5 satellite as shown in Table 2. We used a predefined threshold calculated from the training dataset to detect maneuvers and differentiate them from the normal state without maneuvers. We initially calculated the MDS for all training set samples, then defined the threshold as the 98.8 percentile of the train MDS distribution. The thresholds for USAID and TranAD were set using the Peaks Over Threshold method [21]. For performance metrics, we evaluated the models using TaPR [8], a time series aware Precision (TaP), and Recall (TaR) score because TaP and TaR are shown to be more suitable metrics for time series data. Furthermore, TaP and TaR go beyond direct instance comparison

**Table 1.** Performance evaluation on K3 and K5 satellites for orbit propagation.

Satellite	Methods	Orbit Propagation MSE (SOP Task)						
		Total	$a$	$e$	$i$	$\Omega$	$\omega$	$v$
K3	LR	1.38E-02	1.55E-04	1.25E-02	1.81E-05	2.10E-11	3.43E-02	3.55E-02
	XGBoost	2.55E-03	8.04E-04	1.03E-04	1.23E-02	6.07E-06	1.83E-03	2.83E-04
	LSTM	1.91E-03	1.17E-03	1.40E-04	2.94E-03	3.76E-03	3.01E-03	4.07E-04
	DBN	4.05E-03	1.25E-03	1.41E-03	2.71E-04	1.34E-04	1.49E-02	6.40E-03
	Seq2Seq	1.01E-03	9.58E-04	1.33E-04	1.41E-03	6.68E-04	2.56E-03	3.29E-04
	USAD	1.12E-01	8.66E-02	3.80E-02	9.86E-02	3.07E-01	3.86E-02	1.02E-01
	TranAD	2.49E-03	7.27E-04	4.22E-04	8.91E-04	1.09E-02	4.41E-04	1.52E-03
	GRU-ODE	1.03E-03	2.39E-03	1.66E-04	5.99E-05	1.44E-05	3.08E-03	4.84E-04
	<b>SSPMDA (Ours)</b>	<b>7.15E-04</b>	1.17E-03	1.39E-04	1.41E-04	1.39E-05	2.42E-03	4.08E-04
K5	LR	3.10E-03	7.92E-06	2.06E-03	1.03E-04	4.70E-11	1.22E-02	4.23E-03
	XGBoost	7.92E-04	1.04E-05	3.71E-05	4.17E-03	5.97E-06	4.97E-04	2.98E-05
	LSTM	2.98E-04	3.41E-05	4.96E-05	3.84E-04	2.53E-05	1.24E-03	5.33E-05
	DBN	1.01E-03	6.38E-05	3.15E-04	2.75E-04	3.39E-05	4.99E-03	3.69E-04
	Seq2Seq	2.37E-04	5.31E-05	3.94E-05	4.62E-04	1.09E-04	7.17E-04	3.99E-05
	USAD	2.18E-02	4.58E-04	4.03E-04	1.26E-03	1.28E-01	1.13E-04	2.57E-04
	TranAD	9.68E-04	3.49E-04	1.38E-04	7.19E-04	4.05E-03	1.00E-04	4.48E-04
	GRU-ODE	2.16E-04	2.95E-05	4.77E-05	1.55E-04	5.12E-06	1.01E-03	5.39E-05
	<b>SSPMDA (Ours)</b>	<b>2.14E-04</b>	3.97E-05	4.10E-05	1.17E-04	1.94E-05	1.03E-03	3.54E-05

**Table 2.** Performance evaluation on K5 satellite for maneuver detection. Note that we did not perform maneuver detection on the K3, where there were no maneuvers in 2019.

Methods	Maneuver Detection (SOMD Task)		
	TaP	TaR	AUROC
LR	0.0605	0.7941	0.5091
XGBoost	0.0403	0.8235	0.5036
LSTM	0.0169	0.3824	0.5063
DBN	0.0299	0.7647	0.5048
Seq2Seq	0.0241	0.3824	0.5120
USAD	0.0082	0.5882	0.5188
TranAD	0.0068	0.7353	0.5039
GRU-ODE	0.0521	0.9118	0.5090
<b>SSPMDA (Ours)</b>	<b>0.0963</b>	<b>0.9118</b>	<b>0.5216</b>

by evaluating performance through the consideration of the number of instances. In our work, since missing additional maneuvers could be deadly for satellites, it is more crucial to detect as many instances as possible rather than focusing on their length. Thus, as shown in Table 2, SSPMDA having the highest TaP

and TaR means it misses the fewest maneuvers and hits them most accurately with fewer false alarms. That makes it the most suitable method for SOMD.

We also represented our SSPMDA performance in AUROC, which does not rely on the best threshold. Note that we did not perform maneuver detection on the K3 dataset, where there are no maneuvers in our dataset in 2019. As shown, while most models exhibit high recall, the precision of the models indicates the presence of false positives despite detecting a majority of maneuvers. Also, these results demonstrate that the SOMD task is an extremely challenging one. Another crucial observation is that our SSPMDA achieves the best AUROC performance (0.5216), which is the best differentiation between satellite orbit maneuvers and normal (without maneuvers) operation periods. However, in this experiment, the K5 in 2018 training dataset includes 26 maneuvers, which might have caused a larger boundary to differentiate the maneuvers from the standard operation period, leading to an unsatisfactory AUROC score for all models. For future work, we plan to mitigate this problem by removing the maneuvers in the dataset and training the model with a new fully maneuver-free satellite element dataset, where we can focus on training the model to learn the boundaries of normal orbital behavior better.

## 4.2 Ablation Study

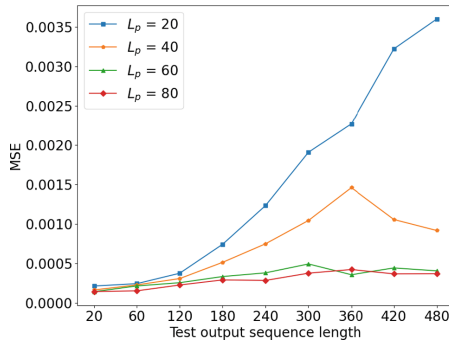
In this work, skip connections are utilized to enhance the performance of the SSPMDA for SOMD [27]. To achieve this, both LC and SC are utilized. The LC is applied to the internal forward part of the NCDEs to perform concatenation operations, while the SC performs element-wise addition within the vector field during the forward pass. We employed both LC and SC to improve the performance. We conducted an ablation study to compare and test the performance of LC and SC. Table 3 presents the AUROC, TaP, and TaR results for the K5 datasets. As shown, AUROC, TaP, and TaR demonstrate the highest performance when both LC and SC are present.

**Table 3.** An ablation study according to the presence or absence of long skip connections (LC) and short skip connections (SC), where the best results are shown in bold. And, ‘w/o’ denotes ‘without’ the respective skip connection.

Methods	TaP	TaR	AUROC
w/o LC, SC	1.0000	0.0442	0.5023
w/o LC	0.7353	0.0999	0.5045
w/o SC	0.7647	0.0562	0.5112
Ours	0.9118	0.0963	<b>0.5216</b>

The communication time with K3 and K5 satellites is extremely limited due to their approximately 90-minute orbital period. Hence, to predict satellite orbits, we used short input sequences. We also employed the same length

for both  $L_p$  and  $L_f$ , as differences in the lengths of the two sequences could potentially degrade the model’s predictive performance. For example, SSPMDA with an input of  $L_p = 20$  might experience slight performance degradation when predicting  $L_f = 480$ . Hence, we conducted an ablation study to demonstrate the trade-off between the performance of dynamic  $L_p$  and  $L_f$ . We evaluated the MSE of SSPMDA with  $L_p$  sequences [20, 40, 60, 80] and  $L_f$  sequences [20, 60, 120, 180, 240, 300, 360, 420, 480] in min. The reason for experimenting with  $L_p$  from 20 to 80 is to ensure that it does not exceed the actual 90-min communication time with the satellite. For  $L_f$ , we set 60-min intervals for performance comparison of satellite propagation, when various  $L_p$  values are used as an input. The experimental setup was identical to previous ones, except for  $L_p$  and  $L_f$ . As shown in Fig. 3, when  $L_p$  is 20 and 40, the overall MSE increases proportionally to  $L_f$ . This indicates that the model is more sensitive to  $L_f$ , when the  $L_p$  length is shorter. In this case, because the complexity of optimization increases as  $L_f$  grows, setting  $L_f$  carefully is essential. Long-term orbit propagation is considerably more complex than short-term orbit propagation. Conversely, as  $L_p$  increases, performance remains stable across all  $L_f$ , indicating that SSPMDA becomes more robust with a longer  $L_p$ . This demonstrates that, as NCDE assumes continuous time, the model is indeed robust to optimization hyperparameters [10, 13]. However, due to the limitations in the calculation speed of NCDEs, setting  $L_p$  too large might not be the best choice [13]. Furthermore, a trade-off between forecast accuracy and prediction length is acceptable in real-world scenarios, where K3 and K5 navigate only about 10 min, three to four times a day.

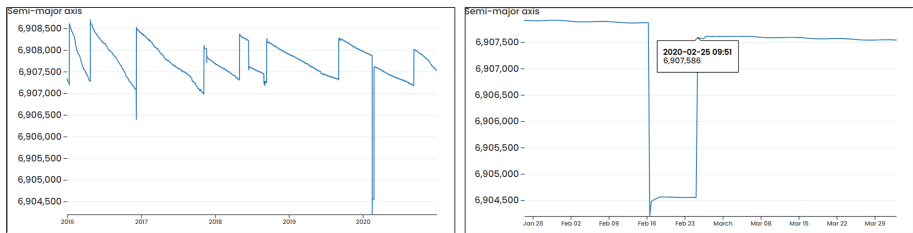


**Fig. 3.** The MSE associated with various lengths of training input sequences  $L_p$  and test prediction sequence length  $L_f$ . Each color represents the  $L_p$  of the training task, where the X-axis is the  $L_f$  of the test task, and the Y-axis represents the MSE score.

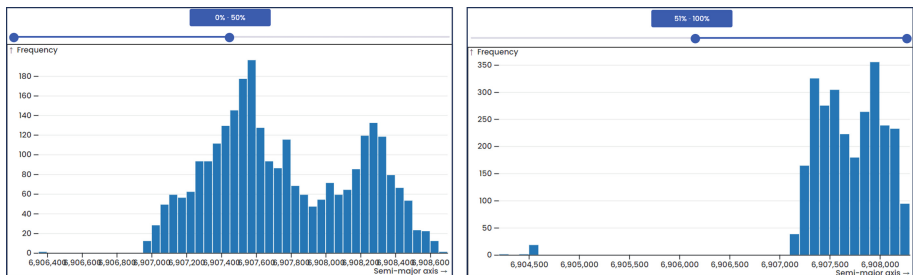
## 5 Visualization

To ensure the explainability of the data and gain insights into the K3 and K5 multipurpose satellites, we developed a visualization analysis tool that can be

deployed and used for real-world satellite operations. In addition to K3 and K5, we utilize the dataset from KOMPSAT-3A (K3A). We use precise orbital elements of the multipurpose satellites from January 2016 to December 2020 for visualization. We developed our explainability tool to provide easy and seamless access through web API calls. Therefore, we compress our data at intervals of 8 hours and 13 minutes and adjust it to have three observations per day. Furthermore, to consider the various environments during satellite orbits, we use the TLE set, where TLE encodes the orbital elements by calculating the satellite's orbital state vectors using simplified perturbation models such as SGP4. We implement an explainability tool using D3.js for this dataset. In addition, our tool is configured to support seven user interactive interfaces: 1) select satellites, 2) select features, 3) adjust the period of the time series, 4) view correlation, 5) check the tooltip over line plot with visualized features, 6) check the connected values by brushing the horizontal axis of the brushable parallel coordinate in which the features are visualized (See Fig. 6).



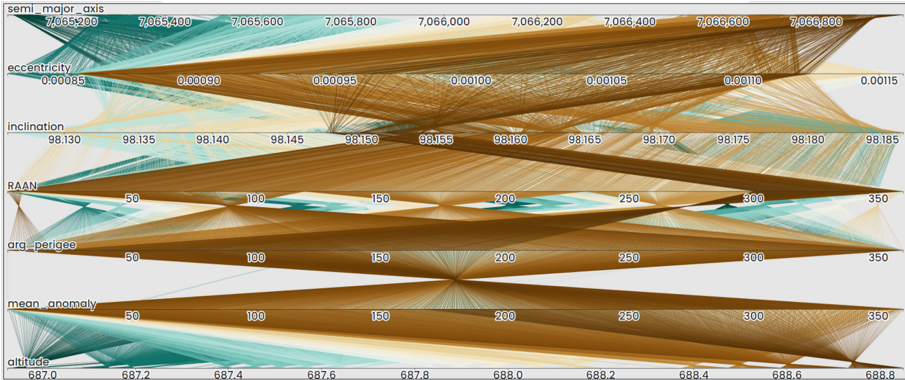
**Fig. 4.** The blue line represents the semi-major axis of K3A, where the left is the plot for all time series, and right is the plot for the selected period. (Color figure online)



**Fig. 5.** This histogram is the semi-major axis of K3A, where the left histogram belongs to the first half of the total time series, and the right belongs to the second half.

The line plot depicted in Fig. 4, a common tool for visualizing time series data, allows us to examine feature values over time when studying satellite





**Fig. 6.** This brushable parallel coordinates plot axis of K3, where the above shows a plot of all data, and the turquoise line on the left side of the lines between the argument of perigee and mean anomaly features is identified as an outlier.

orbital elements. We utilize seven different colors to facilitate easy identification of these features. These plots can effectively depict the time series over the desired period, and we can adjust the period as needed for each satellite. Moreover, when hovering the mouse over the line plot, the tooltip appears, providing the corresponding date and feature value for the selected point. This feature enables a closer examination of specific data points of interest. By observing the line plot as a whole, one can quickly grasp the overall trend and identify the distinct patterns exhibited by each satellite’s orbital elements. Furthermore, within Fig. 4, we find a section where the value suddenly changes while maintaining a certain pattern. To view this interval in detail, it can find the timestamp and value of the section by adjusting the corresponding period, as shown in Fig. 4. The histogram shown in Fig. 5 is a plot to analyze and explain the distribution of orbit features for each satellite and to compare the distribution model with the range. Each distribution is displayed in a different color according to its features, which should be consistent with the line plot. By displaying the distribution of data, it is possible to determine the period during which the average changes based on the difference in the value of each satellite and to observe how well the distribution model is maintained over time. We constructed a histogram for the K3 satellite and analyzed the entire time series data set, dividing it into two equal parts. The results revealed that the overall value distribution remained relatively unchanged, regardless of the duration of the period under consideration. Subsequently, we applied the same analytical approach to the K3A satellite. For example, we can observe notable differences in the distributions of most features, except for the right ascension of the ascending node, across different periods.

Lastly, the brushable parallel coordinates plot shown in Fig. 6 can render the correlation of satellite orbit data. Each feature of the satellite is represented by a horizontal axis, and data points are displayed by connecting lines along the feature axis. By utilizing this visualization, it becomes possible to compare various

ranges and even different units of each feature within a multivariate time series. As a result, the correlation between features can be observed and analyzed. By examining the data, satellite operators can quickly grasp the shape of the changes based on the feature values. It is also possible to select specific feature ranges of interest, focus on particular areas, and identify outliers. In Fig. 6, for example, the argument of perigee and mean anomaly exhibit an inverse relationship, but outliers are identified when brushing certain data points. Therefore, our visualization method in SSPMDA can better capture, explain, and be used to aid in improved analysis on SOP and SOMD tasks.

## 6 Conclusion

In this work, we propose an end-to-end satellite orbit state prediction framework, called Satellite State Prediction and Maneuver Detection Analysis Framework (SSPMDA). SSPMDA is an NCDEs-based framework that combines neural networks and differential equations to effectively predict satellite orbit movement and detect maneuvers. Our experimental results demonstrate that our data-driven modeling can effectively improve satellites' orbital forecasting, compared to other forecasting and detecting maneuver baseline models. Although K3 and K5 perform missions in different orbits and have distinct specifications such as diameter, power and launch mass, resulting in different time series patterns, our model has proven stable performance across both satellites. This confirms that a single generalized model can be applied. We envision that our framework can be deployed and used in real-world satellite operations and support other countries' satellite orbit state predictions.

**Acknowledgments.** We thank Korea Aerospace Research Institute (KARI) for supporting this research. Also, we would like to give sincere and special thanks to Jinbeom Kim and Sangyup Lee for running initial experiments, and helping and drafting the initial and earlier version of this work, while they were at Sungkyunkwan University. In addition, this work was partly supported by Institute for Information & communication Technology Planning & evaluation (IITP) grants funded by the Korean government MSIT: (RS-2022-II221199, RS-2024-00337703, RS-2022-II220688, RS-2019-II190421, RS-2023-00230337, RS-2024-00356293, RS-2022-II221045, and RS-2021-II212068).

## References





1. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 3395–3404 (2020)
2. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Sons (2016)
3. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in neural information processing systems **31** (2018)

4. Ciccone, M., Gallieri, M., Masci, J., Osendorfer, C., Gomez, F.: Nais-net: Stable deep networks from non-autonomous differential equations. *Advances in Neural Information Processing Systems* **31** (2018)
5. De Brouwer, E., Simm, J., Arany, A., Moreau, Y.: Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems* **32** (2019)
6. Dormand, J.R., Prince, P.J.: A family of embedded runge-kutta formulae. *J. Comput. Appl. Math.* **6**(1), 19–26 (1980)
7. Han, S., Woo, S.S.: Learning sparse latent graph representations for anomaly detection in multivariate time series. In: *Proceedings of the 28th ACM SIGKDD Conference on knowledge discovery and data mining*. pp. 2977–2986 (2022)
8. Hwang, W.S., Yun, J.H., Kim, J., Kim, H.C.: Time-series aware precision and recall for anomaly detection: Considering variety of detection result and addressing ambiguous labeling. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. pp. 2241–2244. ACM (2019)
9. Institute, K.A.R.: The people in the control room who track satellites 24 hours a day (2017), <https://www.segye.com/print/20170519003341>
10. Jhin, S.Y., Lee, J., Park, N.: Precursor-of-anomaly detection for irregular time series. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. pp. 917–929 (2023)
11. Jhin, S.Y., Shin, H., Hong, S., Jo, M., Park, S., Park, N., Lee, S., Maeng, H., Jeon, S.: Attentive neural controlled differential equations for time-series classification and forecasting. In: *2021 IEEE International Conference on Data Mining (ICDM)*. pp. 250–259 (2021). <https://doi.org/10.1109/ICDM51629.2021.00035>
12. KARI: KOMPSAT-3,5 Multipurpose Satellites, <https://www.kari.re.kr/eng/sub03.03.do>
13. Kidger, P., Morrill, J., Foster, J., Lyons, T.: Neural controlled differential equations for irregular time series. *Adv. Neural. Inf. Process. Syst.* **33**, 6696–6707 (2020)
14. Kim, Y.G., Yun, J.H., Han, S., Kim, H.C., Woo, S.S.: Revitalizing self-organizing map: Anomaly detection using forecasting error patterns. In: *IFIP International Conference on ICT Systems Security and Privacy Protection*. pp. 382–397. Springer (2021)
15. Melvin, P.J.: A kalman filter for orbit determination with applications to gps and stellar navigation. *Spaceflight mechanics* **1996**, 719–738 (1996)
16. Pearlman, M., Degnan, J., Bosworth, J.: The international laser ranging service. *Adv. Space Res.* **30**(2), 135–143 (2002)
17. Peng, H., Bai, X.: Machine learning approach to improve satellite orbit prediction accuracy using publicly available data. *J. Astronaut. Sci.* **67**(2), 762–793 (2020)
18. Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., Zhang, Q.: Time-series anomaly detection service at microsoft. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. pp. 3009–3017 (2019)
19. Rubanova, Y., Chen, R.T., Duvenaud, D.K.: Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems* **32** (2019)
20. Shin, Y., Lee, S., Tariq, S., Lee, M.S., Jung, O., Chung, D., Woo, S.S.: Itad: Integrative tensor-based anomaly detection system for reducing false positives of satellite systems. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. pp. 2733–2740 (2020)

21. Siffer, A., Fouque, P.A., Termier, A., Largouet, C.: Anomaly detection in streams with extreme value theory. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1067–1075 (2017)
22. Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint [arXiv:2201.07284](https://arxiv.org/abs/2201.07284) (2022)
23. Vittaldev, V.: Uncertainty propagation and conjunction assessment for resident space objects (2015), <https://repositories.lib.utexas.edu/handle/2152/32906>, accessed on 11.25.2021
24. Woo, S.S., Yoon, D., Gim, Y., Park, E.: Raad: Reinforced adversarial anomaly detector. In: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing. pp. 883–891 (2024)
25. Yoon, D., Woo, S.S.: Who is delivering my food? detecting food delivery abusers using variational reward inference networks. In: Proceedings of the 29th ACM international conference on information & knowledge management. pp. 2917–2924 (2020)
26. Yun, J.H., Kim, J., Hwang, W.S., Kim, Y.G., Woo, S.S., Min, B.G.: Residual size is not enough for anomaly detection: improving detection performance using residual similarity in multivariate time series. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. pp. 87–96 (2022)
27. Zhu, Q., Shen, Y., Li, D., Lin, W.: Neural piecewise-constant delay differential equations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 9242–9250 (2022)



# MIXAD: Memory-Induced Explainable Time Series Anomaly Detection

Minha Kim<sup>1</sup> , Kishor Kumar Bhaumik<sup>1,2</sup> , Amin Ahsan Ali<sup>2</sup>,  
and Simon S. Woo<sup>1</sup>  

<sup>1</sup> Sungkyunkwan University, Suwon, Republic of Korea  
{sunshine01,kishor25,woo}@g.skku.edu

<sup>2</sup> Center for Computational and Data Sciences, Independent University,  
Dhaka, Bangladesh  
aminali@iub.edu.bd

**Abstract.** For modern industrial applications, accurately detecting and diagnosing anomalies in multivariate time series data is essential. Despite such need, most state-of-the-art methods often prioritize detection performance over model interpretability. Addressing this gap, we introduce *MIXAD* (Memory-Induced Explainable Time Series Anomaly Detection), a model designed for interpretable anomaly detection. *MIXAD* leverages a memory network alongside spatiotemporal processing units to understand the intricate dynamics and topological structures inherent in sensor relationships. We also introduce a novel anomaly scoring method that detects significant shifts in memory activation patterns during anomalies. Our approach not only ensures decent detection performance but also outperforms state-of-the-art baselines by **34.30%** and **34.51%** in interpretability metrics. The code for our model is available at <https://github.com/mhkim9714/MIXAD>.

**Keywords:** Anomaly detection · Explainable AI · Time series

## 1 Introduction

The proliferation of sensors and Internet of Things (IoT) devices in healthcare, smart manufacturing, and cybersecurity has significantly increased the generation of time series data, emphasizing the need for advanced Multivariate Time Series (MTS) analysis [1, 4, 13]. Anomaly detection, crucial for system diagnosis and maintenance, faces challenges due to the rarity of anomalies and the dynamic nature of data. These issues complicate manual labeling and necessitate a deep understanding of both intra-metric (within a single series) and inter-metric (across different series) dependencies. Furthermore, enhancing explainability in MTS anomaly detection is essential for improving operational

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_16](https://doi.org/10.1007/978-3-031-78189-6_16).

clarity and decision-making, particularly in pinpointing and explaining the root causes of anomalies. In environments like smart manufacturing with numerous sensors, clarity about which sensors or interactions between sensors contribute to an anomaly is vital. Despite existing methods' capabilities in detecting anomalies, many lack the ability to provide clear explanations for their findings [12]. Enhancing autoencoder models with an external memory module has emerged as a promising solution. This approach not only boosts detection performance but also deepens insights into the dynamics of anomalies, thereby improving model interpretability and building trust in anomaly detection systems [9].

Addressing gaps in MTS anomaly detection, we introduce *MIXAD*. For the first time in MTS anomaly detection, *MIXAD* incorporates a memory module to store the complex dynamics of data extracted through a spatiotemporal feature extractor. This adoption sets *MIXAD* apart from most baselines, as our model simultaneously models intra-metric and inter-metric dependencies, offering enhanced insight into the nature of the data. Additionally, we generate a novel anomaly score through memory activation pattern analysis, which successfully combines the accuracy of anomaly detection with improved elucidation of root causes. Unlike most existing methods that rely heavily on forecasting or reconstruction errors to construct the anomaly score—typically identifying the feature with the largest error as the cause of the anomaly—our approach employs a more sophisticated scoring method. This method explains which features contribute to the anomaly and how by analyzing the differences in memory activation patterns between normal and abnormal periods. Furthermore, by adopting the Pearson correlation calculation in a post hoc manner, we can further determine which features share similar patterns of memory activation shift. *MIXAD* offers critical, actionable insights for practical applications by blending the precision of unsupervised learning with the transparency of explainable AI. The contributions of this work are outlined as follows:

1. We present *MIXAD*, a pioneering approach that significantly improves the interpretability of anomaly detection outcomes in the MTS domain. This innovation addresses a significant oversight in existing research, making the model applicable across diverse fields.
2. For the first time in MTS anomaly detection, *MIXAD* integrates a memory network designed to capture and retain the normal spatiotemporal patterns of data. We also introduce a novel anomaly scoring mechanism that leverages memory attention pattern matching, maintaining robust detection performance while significantly augmenting the model's interpretive capabilities.
3. We employ benchmark datasets with interpretation labels, enabling thorough evaluation of our model's detection capabilities and interpretive efficacy. *MIXAD* not only achieves robust detection results but also significantly outperforms existing state-of-the-art (SOTA) baselines in interpretability metrics by **34.30%** and **34.51%**.

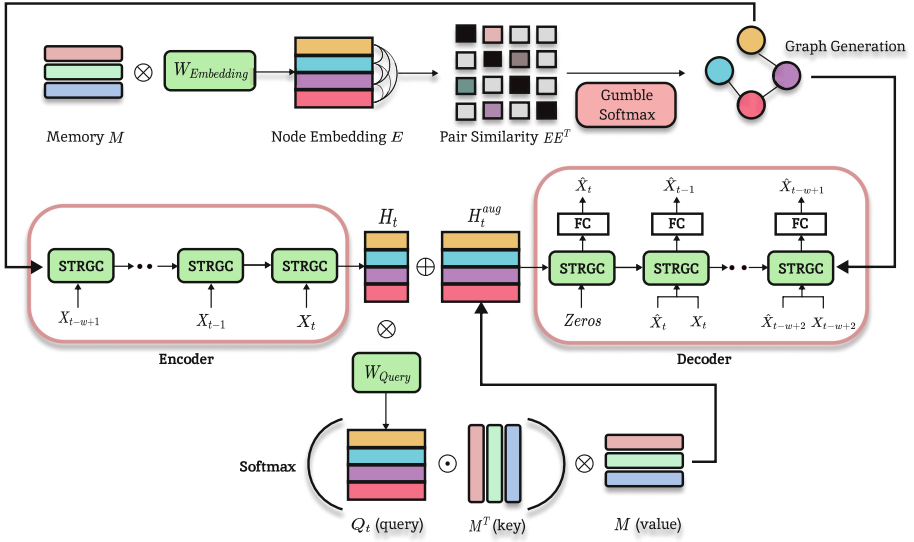
## 2 Related Works

**Time Series Anomaly Detection.** Anomaly detection in large-scale databases has become more challenging with the emergence of diverse data modalities [8,12]. The widespread use of sensors and IoT devices has significantly increased data volumes, underscoring the need for precise anomaly detection in time-series databases. These databases often exhibit stochastic and temporal patterns from various engineering sources, necessitating effective differentiation of outliers [20]. Due to the scarcity of labeled data and the diversity of anomalies [1], the current research trend is biased towards unsupervised learning models. Time-series anomaly detection research is broadly classified into two main categories: univariate models that analyze individual time series and multivariate methods that examine multiple series concurrently. Despite progress, many SOTA methods still lack a quantitative evaluation of model explainability, which is essential for real-world applications. Our work focuses on enhancing the interpretability of models in practical environments. For detailed discussions on recent developments, please refer to the supplementary materials.

**Memory Network for Anomaly Detection.** Recent advancements in anomaly detection have highlighted the effectiveness of memory-augmented attention (MAA) models, particularly for their capacity to store normal data patterns encountered during training. For example, Gong et al. [7] integrated a memory module into an autoencoder to mitigate the model’s tendency to reconstruct anomalies accurately, thus enhancing detection precision using reconstruction errors. Similarly, Park et al. [18] developed an unsupervised video anomaly detection method that utilizes a memory module to record normal data patterns, enhancing the discriminative power of the memory items and the abnormal features. This approach also uses feature compactness and separateness losses to ensure the diversity and discriminative capabilities of memory items, thereby improving anomaly detection efficiency and effectiveness. Inspired by these approaches, we propose the novel integration of a memory module into the MTS anomaly detector to enhance performance, providing deeper insights into how anomalous patterns diverge from normal patterns.

## 3 Method

Figure 1 provides an overview of the *MIXAD* framework. The model features an encoder-decoder architecture, with the Spatiotemporal Recurrent Convolution Unit (STRGC) serving as the core component for both the encoder and the decoder. An external memory assists in augmenting the encoded hidden representations to better initialize the decoder’s hidden states. Additionally, the learned memory is processed through an embedding layer to generate the graph structure utilized within the STRGC. The subsequent subsections will detail each component of *MIXAD*, beginning with an explanation of the basic problem formulation.



**Fig. 1.** Overview of the MIXAD framework: Initially, a sparse graph is constructed by calculating pairwise similarities between memory-based node embeddings. Subsequently, input data is processed through a STRGC-based encoder and decoder for self-reconstruction. Throughout this process, an external memory enhances the encoded feature vector by utilizing an attention mechanism between the original feature vector and the memory.

### 3.1 Problem Formulation

In this study, we introduce a framework for unsupervised anomaly detection, analyzing time series data with  $N$  features over a period  $T$ , denoted as  $X \in \mathbb{R}^{N \times T}$ . For simplicity, we denote feature indices with superscripts and timestamps with subscripts, allowing for the representation of data at any time  $t$  as  $X_t \in \mathbb{R}^{N \times 1}$ . We normalize both training and testing data using feature-wise min-max normalization and employ a sliding window technique with window size  $w$  to capture temporal dependencies, denoted as  $W_t = X_{t-w+1:t}$ . As the traditional reconstruction-based methods, our proposed model,  $f$ , is trained to reconstruct the input window:  $\hat{W}_t = f(W_t)$ . However, MIXAD distinguishes itself from existing methods by adopting a novel anomaly scoring mechanism that utilizes memory activation analysis. The model computes an anomaly score  $s_t$  for each timestamp  $t$  and compares it to a predefined threshold to determine anomalies. Since MIXAD aims to not only detect anomalies but also pinpoint the specific features contributing to anomalies, our main objectives can be summarized as follows:

- **Anomaly Detection:** Determine if  $X_t$  at a given timestamp is anomalous.
- **Anomaly Interpretation:** Identify which features contribute to the anomaly at the identified timestamps.



### 3.2 Spatiotemporal Recurrent Convolution Unit

Drawing inspiration from the successful implementation of temporal modeling with LSTM-based encoder-decoder structures [16], *MIXAD* adopts a similar architecture. The encoder first processes the input time series window, extracting a fixed-length feature vector. Subsequently, the decoder utilizes this representation to reconstruct the time series in reverse order.

Recognizing that MTS exhibit both temporal and spatial dependencies among features, recent advancements in GNNs have led to a notable development of STRGC, which incorporates graph convolution operations into recurrent cells [11, 24]. In our work, by substituting LSTM with STRGC, *MIXAD* concurrently processes all series, capturing the data’s spatial and temporal dependencies more effectively. The graph convolution operation is defined as follows:

$$W_{*A}X_t = \sum_{k=0}^K \tilde{A}^k X_t W_k \quad (1)$$

where  $*A$  represents a graph convolution operation with input  $X_t \in \mathbb{R}^{N \times 1}$  and kernel parameters  $W \in \mathbb{R}^{K \times 1 \times h}$ , approximated using Chebyshev polynomials to the order of  $K$  [5]. This operation requires an adjacency matrix  $A \in \mathbb{R}^{N \times N}$ , normalized to  $\tilde{A}$ , to outline the data’s topological structure. Building on this, the STRGC-enhanced GRU cell updates are as follows:

$$\begin{aligned} r_t &= \text{sigmoid}(W_{r*A}[X_t \parallel H_{t-1}] + b_r) \\ u_t &= \text{sigmoid}(W_{u*A}[X_t \parallel H_{t-1}] + b_u) \\ C_t &= \text{tanh}(W_{C*A}[X_t \parallel (r_t \odot H_{t-1})] + b_C) \\ H_t &= u_t \odot H_{t-1} + (1 - u_t) \odot C_t \end{aligned} \quad (2)$$

where  $r$ ,  $u$ , and  $C$  denote the reset gate, update gate, and candidate state within the GRU cell, respectively, with  $W_{\{r,u,C\}} \in \mathbb{R}^{K \times (1+h) \times h}$  and  $b_{\{r,u,C\}} \in \mathbb{R}^h$  representing the gate parameters. The hidden representation produced by STRGC is denoted by  $H_t \in \mathbb{R}^{N \times h}$ . This adaptation enables our model to leverage graph convolution within a GRU cell, providing a sophisticated method to dissect spatiotemporal dependencies in multivariate time series data.

### 3.3 Memory-Augmented Graph Structure Learning

In the realm of spatiotemporal modeling, tasks like traffic forecasting leverage spatial adjacency information, often readily available from urban maps [24]. However, applying similar spatial correlation concepts to MTS anomaly detection presents challenges. In MTS anomaly detection, identifying dependencies among features is not immediately obvious and typically requires domain expertise to formalize these correlations. To overcome this obstacle, our research incorporates a Graph Structure Learning (GSL) method. GSL excels at systematically discovering and applying spatial correlations among features without prior explicit

knowledge, thereby significantly enhancing the anomaly detection capabilities of models dealing with complex MTS data.

To streamline our examination of GSL within the context of MTS anomaly detection, we highlight its application in the Graph Deviation Network (GDN) framework [6]. GDN employs learnable embeddings for each feature, establishing feature connections based on embedding similarity. While effective, this method creates a static graph that may not fully accommodate the dynamic nature of non-stationary time series data. To address these limitations and accommodate the temporal evolution of data, we employ a dynamic graph learning strategy introduced in Structure Learning Convolution (SLC) [25]. SLC utilizes input-conditioned node embeddings, creating a graph that dynamically adapts over time, thus providing a more flexible and responsive modeling of feature relationships. This method enhances our model’s ability to adjust to changing data patterns, although challenges remain in preventing over-sensitivity to immediate data variations and ensuring consistency in the graph structure.

To address the aforementioned limitations of traditional GSL methods, we draw from foundational research [11] to enhance spatiotemporal graph learning with a memory module. In *MIXAD*, each memory item is updated at each iteration, storing node-level spatiotemporal prototypes that generate node embeddings adaptable to non-stationary time series without being overly sensitive to noisy data. The memory module, denoted as  $M \in \mathbb{R}^{m \times d}$ , where  $m$  represents the number of memory items and  $d$  the dimension of each item, plays a pivotal role in pattern recognition and modeling feature relationships. The interaction with the memory module is captured through:

$$\begin{aligned} Q_t &= H_t * W_Q + b_Q \\ Att_t &= \text{softmax}(Q_t * M^T) \\ H_t^{aug} &= Att_t * M \end{aligned} \tag{3}$$

Here,  $H_t^{aug} \in \mathbb{R}^{N \times d}$  represents a memory-augmented hidden state derived via an attention mechanism [22]. The process begins by transforming the hidden representations  $H_t$  into a query space, resulting in  $Q_t \in \mathbb{R}^{N \times d}$ , using the parameters  $W_Q \in \mathbb{R}^{h \times d}$  and  $b_Q \in \mathbb{R}^d$ . The memory module  $M$  is then utilized as both key and value in the attention operation, allowing each feature’s query vector to compute similarity scores across  $m$  memory items. These attention scores  $Att_t \in \mathbb{R}^{N \times m}$  generate an augmented hidden state  $H_t^{aug}$  through a weighted aggregation of memory items, enhancing the initial hidden representations. This augmented state is subsequently concatenated with  $H_t$ , initializing the decoder’s hidden state. This approach not only enriches the model’s capacity to discern complex feature relationships but also adapts to their dynamism over time.

In our spatiotemporal graph learning framework, the memory module  $M$  is essential for creating advanced node embeddings that are crucial for structuring the input for the STRGC encoder and decoder. However, we encountered a challenge with the graph adjacency matrix becoming overly dense, cluttered with unnecessary edges that could obscure the spatial relationships between features. To address this issue, we introduced a regularized graph generation mod-

ule aimed at promoting sparsity in the adjacency matrix. This ensures that the graph structure more accurately captures the essential feature connections [24]. The regularized graph generation process can be expressed as follows:

$$\begin{aligned} \theta &= (W_E * M)(W_E * M)^T \\ A &= \sigma((\log(\theta^{ij}/(1 - \theta^{ij})) + (g1^{ij} - g2^{ij}))/\tau) \\ \text{s.t. } g1^{ij}, g2^{ij} &\sim \text{Gumbel}(0, 1) \end{aligned} \quad (4)$$

where  $W_E \in \mathbb{R}^{N \times m}$  projects the memory onto an embedding space, resulting in a probability matrix  $\theta \in \mathbb{R}^{N \times N}$ . Each element  $\theta^{ij}$  reflects the potential for an edge between features  $i$  and  $j$ . Utilizing the Gumbel softmax technique, we convert  $\theta$  into a discretely sparse adjacency matrix  $A \in \mathbb{R}^{N \times N}$ , maintaining differentiability for gradient-based optimization. In Eq. (4),  $\sigma$  represents the activation function and  $\tau$  the softmax temperature.

### 3.4 Memory-Induced Explainable Anomaly Detection (MIXAD)

Our proposed *MIXAD* architecture illustrated in Fig. 1, employs the STRGC framework, enhanced with a memory module, to classify nodes with similar spatiotemporal dynamics into specific memory slots. This setup aims to efficiently identify and store distinct patterns within each memory item, improving the model’s discriminative power. To refine this capability, we incorporate three specialized loss functions, as follows:

$$\begin{aligned} L_1 &= \sum_{t,i}^{T,N} \max\{\|Q_t^i - M[pos_t^i]\|^2 - \|Q_t^i - M[neg_t^i]\|^2 + \lambda, 0\} \\ L_2 &= \sum_{t,i}^{T,N} \|Q_t^i - M[pos_t^i]\|^2 \\ L_3 &= -\log(m) - \frac{1}{m} \sum_{j=1}^m \log\left(\frac{\exp(\sum_{t,i}^{T,N} Att_t^{ij})}{\sum_{k=1}^m \exp(\sum_{t,i}^{T,N} Att_t^{ik})}\right) \\ \text{Loss} &= L_{MAE} + \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 \end{aligned} \quad (5)$$

The core of our method leverages  $Q_t^i \in \mathbb{R}^d$  as an anchor to identify the nearest ( $M[pos_t^i] \in \mathbb{R}^d$ ) and the second-nearest memory items ( $M[neg_t^i] \in \mathbb{R}^d$ ) based on attention scores. We use a triplet margin loss  $L_1$  to ensure a significant margin between the closest and second-closest slots for improved separation, while a compact loss  $L_2$  maintains close proximity between the closest slot and the anchor. To mitigate training instabilities and uneven memory utilization, we also implement a Kullback-Leibler (KL) divergence loss  $L_3$ , which promotes a uniform attention distribution across  $m$  memory items. The overall training objective, as presented in the bottom of Equation (5), combines these losses with a Mean Absolute Error (MAE)-based reconstruction loss, regulated by balancing parameters  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ . This integrated objective stabilizes the training

process and enhances the interpretability and efficiency of memory items in our anomaly detection model. We selected MAE as the most suitable reconstruction loss over Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) after considering their characteristics. In unsupervised anomaly detection, normal instances are typically used for training, but high-purity normal datasets are rare in real-world data, making it crucial to minimize the influence of noise. MSE and RMSE are significantly affected by noise due to squaring the differences between predicted and ground truth values, leading to training instability. Therefore, we adopted the more robust MAE loss.

### 3.5 Anomaly Scoring

Traditional reconstruction-based anomaly detectors utilize the reconstruction error to generate an anomaly score, as shown by the equation  $s_t = |W_t - \hat{W}_t|$ . However, advanced models often reconstruct anomalies too accurately, thus diminishing the effectiveness of this approach. Furthermore, the presence of contextual anomalies, where individual series distributions appear normal, but their interrelations are anomalous, necessitates more sophisticated scoring methods.

To address these challenges, we introduce a novel anomaly scoring method that leverages memory query attention scores. This method assumes that inputs deviating from learned normal patterns will significantly alter the distribution of attention scores for the memory components. The anomaly score for timestamp  $t$ ,  $s_t \in \mathbb{R}^N$ , is calculated by evaluating the shift in attention score distributions between consecutive timestamps, using the Jensen-Shannon divergence (JSD):

$$s_t = [JSD(Att_{t-1}^i \parallel Att_t^i)]_{i=1}^N \quad (6)$$

Subtle differences in memory activation patterns, even when the reconstruction error is not notably large or in the presence of contextual anomalies, enable the model to identify anomalies more accurately. Furthermore, understanding how memory activation changes enhances our ability to interpret the detected anomalies deeply. For example, if a node that typically references memory item 1 shifts to referencing memory item 2 at a certain timestamp, obtaining this information provides a deeper insight into the nature of the anomaly. Meanwhile, for time series that display distinct cyclic temporal patterns, normal seasonal fluctuations might inadvertently affect the anomaly score  $s \in \mathbb{R}^{T \times N}$ , complicating anomaly detection. To mitigate this, we apply Seasonal-Trend decomposition using LOESS (STL) [3] on the anomaly scores and remove identifiable seasonal components based on the period  $P$ , determined from the Real Fast Fourier Transform (RFFT) analysis [19] of  $s^i$  for each feature  $i$ . The process is defined as:

$$\forall i \in \{1, \dots, N\}, s^i = s^i - STL\{s^i; P\}_{seasonal} \quad (7)$$

$$P = \left( \frac{2\pi}{\Delta t \cdot \operatorname{argmax}(|\mathcal{F}\{s^i\}|)} \right)$$

Using the dominant frequency of signal  $s^i$  identified by the maximum modulus of the FFT ( $\mathcal{F}$ ) output,  $P$  is calculated from the RFFT frequency spectrum, with

$\Delta t$  as the sampling interval. The seasonal component is extracted using STL, and subsequent deseasonalization yields a refined anomaly score  $s' \in \mathbb{R}^{T \times N}$ . To determine the overall anomaly level at each timestamp  $t$ , we aggregate  $s'_t$  across all features using the max function. A timestamp  $t$  is flagged as anomalous if its aggregated score surpasses a set threshold.

### 3.6 Anomaly Interpretation

Anomaly Interpretation is essential in anomaly detection, yet it has been insufficiently addressed in much of the recent research [2, 15]. The ability to interpret and trust the outputs of deep learning models is crucial, especially as these models are increasingly utilized to analyze datasets in various real-world applications. Traditionally, identifying the contributing factors of detected anomalies has depended on analyzing the reconstruction or prediction error for each data dimension. However, this approach hinges on the model’s accuracy, potentially compromising interpretability when models inaccurately handle input data. Our examination of existing literature reveals a notable gap in quantitatively measured interpretability. To bridge this gap and enhance the utility of anomaly detection, we introduce a new method for anomaly interpretation in our study.

For each detected anomalous segment, we observe that the anomaly scores of ground truth causal features often exhibit similar patterns that set them apart from non-causal features. To pinpoint the set of causal features of an anomaly, we begin by identifying the feature with the highest anomaly score in the segment. We then calculate the Pearson correlation coefficient between the anomaly scores of every feature and the anomaly score of the identified feature. Ranking the features based on the absolute values of their correlation coefficients allows us to order them from most to least likely to have caused the anomaly. This approach provides a detailed and interpretable method for analyzing anomalies, improving the model’s utility and reliability in practical applications.

## 4 Experiments and Analysis

### 4.1 Datasets and Baselines

**Datasets.** We utilized two publicly available datasets, the Server Machine Dataset (SMD) [20] and the Multi-Source Distributed System (MSDS) Dataset [17], specifically chosen for their availability of ground truth information on anomalous features within the datasets. These additional interpretation labels enable us to quantitatively evaluate and compare the effectiveness of our interpretative approach against the baselines. Detailed information about these datasets is provided in the supplementary materials.

**Baselines.** We compared a range of selected SOTA anomaly detection algorithms, chosen for their explicit emphasis on enhancing the interpretability of detection results, against our *MIXAD*. The algorithms, MTAD-GAT [26], GDN [6], TranAD [21], DuoGAT [14], and DAEMON [2], are notable for their

advanced methods in elucidating complex data interactions and enhancing detection accuracy. For a comprehensive description of each baseline algorithm, please refer to the supplementary materials.

### 4.2 Evaluation Metrics

To evaluate MIXAD’s effectiveness against competing models, we use precision, recall, and the F1-score, incorporating a point-adjusted evaluation method for a more realistic assessment of anomaly detection [20,23]. This method acknowledges that real-world anomalies typically span multiple timestamps, treating the identification of any part of an anomaly segment as a correct detection. For anomaly interpretation, MIXAD’s ability to detect actual anomalous features among its top predictions is measured using the HitRate@P% metric. This metric adjusts the evaluation scope based on the proportion of ground truth features, offering a nuanced understanding of the model’s diagnostic accuracy. The formula for calculating HitRate@P% is as follows:

$$HitRate@P\% = \frac{Hit@[P\% \times |GT|]}{|GT|} \tag{8}$$

where  $|GT|$  denotes the number of ground truth causal features, and  $P\%$  represents the percentage of ground truth dimensions evaluated at each timestamp. Following prior works [21], we use 100 and 150 for  $P$ .

**Table 1.** Performance comparison of MIXAD with baseline models on the SMD and MSDS datasets. The highest performance according to the F1-score and HitRate@P% is highlighted in bold, while the second-best performance is underlined.

Dataset	Method	Precision	Recall	F1	HitRate@100%	HitRate@150%
SMD	MTAD-GAT	0.8889	0.7943	0.8318	0.3716	0.4801
	GDN	0.9114	0.8917	0.8968	0.2994	0.4285
	TranAD	0.9595	0.9325	0.9446	0.3628	0.4747
	DuoGAT	0.9924	0.9945	<b>0.9965</b>	<u>0.3825</u>	<u>0.5155</u>
	DAEMON	0.9456	0.9746	0.9595	0.3304	0.4574
<b>MIXAD</b>		0.9703	0.9884	<u>0.9792</u>	<b>0.5137</b>	<b>0.6672</b>
MSDS	MTAD-GAT	0.9919	0.7964	0.8835	<u>0.5812</u>	0.5885
	GDN	0.9989	0.8026	0.8900	0.2276	0.3382
	TranAD	0.9859	0.9749	<b>0.9804</b>	0.4583	0.6253
	DuoGAT	0.9634	0.9576	0.9605	0.4435	<u>0.6614</u>
	DAEMON	0.9711	0.9450	0.9578	0.3358	0.5115
<b>MIXAD</b>		0.9716	0.9540	<u>0.9627</u>	<b>0.7818</b>	<b>0.8136</b>

### 4.3 Performance Comparisons

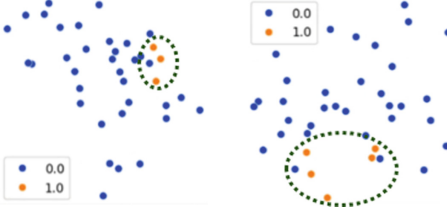
We thoroughly evaluated *MIXAD*'s performance through comprehensive experiments, with setup details provided in the materials. Utilizing a grid search method, we determined the optimal anomaly thresholds for each experiment based on the highest F1-scores [20]. As shown in Table 1, *MIXAD* achieved competitive detection accuracy for both the SMD and MSDS datasets, closely matching SOTA models. Notably, while its detection accuracy in the SMD dataset ranked second, slightly below the top SOTA model by 1.73% and above the second-best by 2.05%, *MIXAD* significantly outperformed all models in interpretability. It improved interpretation scores by **34.30%** and **29.43%** in HitRate@100% and HitRate@150%, respectively, surpassing the previously highest-ranking DuoGAT algorithm. In the MSDS dataset, *MIXAD* again showed detection performance slightly below the best by 1.81%, yet it improved upon the second highest by 0.23%, placing it second overall. More impressively, it raised the bar for interpretability, setting new records with increases of **34.51%** and **23.01%** in HitRate@100% and HitRate@150%, respectively. These highlight *MIXAD*'s strong detection abilities and its superior interpretative performance, demonstrating its potential for real-world applications (Table 2).

**Table 2.** Ablation study on the SMD dataset. The highest performance according to the F1-score and HitRate@P% is highlighted in bold.

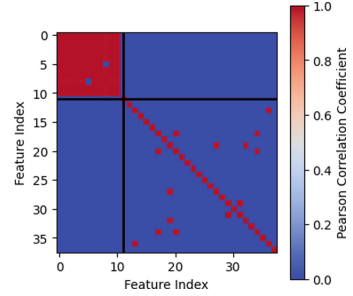
Methods	Precision	Recall	F1	HitRate@100%	HitRate@150%
- Reconstruction	0.9410	0.9864	0.9629	0.4130	0.5562
- $L_3$	0.9617	0.9719	0.9667	0.3537	0.4868
- Memory module	0.9793	0.9493	0.9626	0.4317	0.5695
- New anomaly score	0.9767	0.9506	0.9624	0.4278	0.5658
- New interpretation	0.9703	0.9884	<b>0.9792</b>	0.4829	0.5970
<b><i>MIXAD</i></b>	0.9703	0.9884	<b>0.9792</b>	<b>0.5137</b>	<b>0.6672</b>

### 4.4 Ablation Study

Our ablation study aimed to evaluate the individual contributions of various components within *MIXAD* towards its detection and interpretation capabilities. The study involved several modifications: (1) replacing the reconstruction-focused decoder with a forecasting one, (2) removing the KL divergence loss ( $L_3$ ) which promotes uniform memory activation, (3) substituting the memory module ( $M$ ) with a learnable node embedding similar to the GDN baseline [6], (4) discarding the newly proposed anomaly scoring method and instead relying solely on reconstruction error for scoring, and (5) excluding our novel interpretation method based on Pearson correlation among anomaly scores.



**Fig. 2.** T-SNE visualization of node embeddings from two anomaly segments of the SMD dataset.



**Fig. 3.** Heatmap visualization of Pearson correlation coefficients for anomaly scores.

The results are detailed in Table 3. The findings highlight the integral role each component plays in *MIXAD*'s performance. Notably, the exclusion of any key component diminished both detection and interpretability, illustrating their collective importance. However, the omission of the novel interpretation method did not affect detection accuracy, as it employed the same enhanced anomaly scoring mechanism  $s'$ . Our analysis indicates that while all components significantly enhance interpretability, the KL divergence loss ( $L_3$ ) is particularly effective. It prevents biased learning that could lead to the underutilization of some memory slots, thereby markedly improving model explainability. Additionally, the proposed post hoc interpretation technique plays a complementary role, further refining the final interpretability of the detection output.

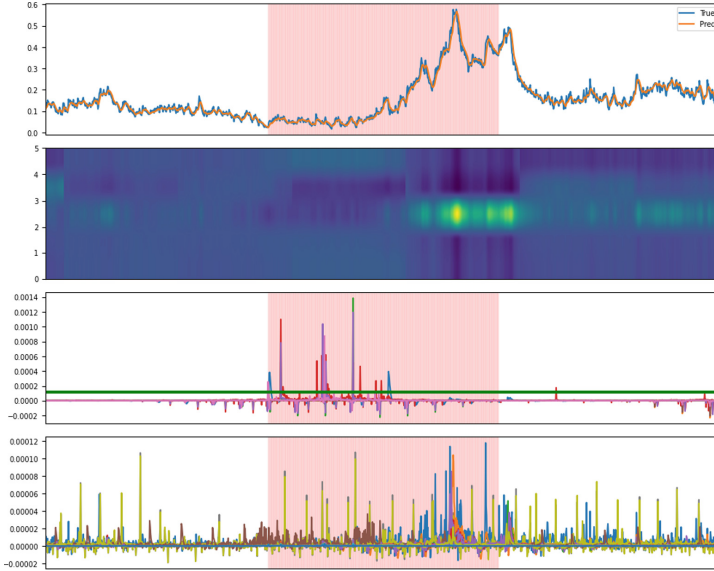
#### 4.5 Visualization of Node Embeddings

We conducted a qualitative assessment of the node embeddings' quality by employing t-SNE to visualize them in a low-dimensional space. As depicted in Fig. 2, the orange and blue points represent the root cause and non-causal features of two anomalous segments within the SMD dataset, respectively. Consistent with our expectations, the visualization reveals that the memory-based node embeddings form distinct clusters based on feature relationships and their contributions to anomalies. This clustering demonstrates the memory module's ability to capture and retain each feature's unique spatiotemporal attributes, allowing for the creation of analogous embeddings for similar nodes. Such capability not only aids in accurate time series reconstruction but also significantly boosts the model's effectiveness in anomaly detection and interpretation, showcasing the memory's integral role in improving *MIXAD*'s functionality.

#### 4.6 Visualization of Anomaly Scores

To demonstrate the efficacy of *MIXAD*'s innovative anomaly scoring method, we visualized an anomalous segment from the SMD dataset, depicted in Fig. 4.





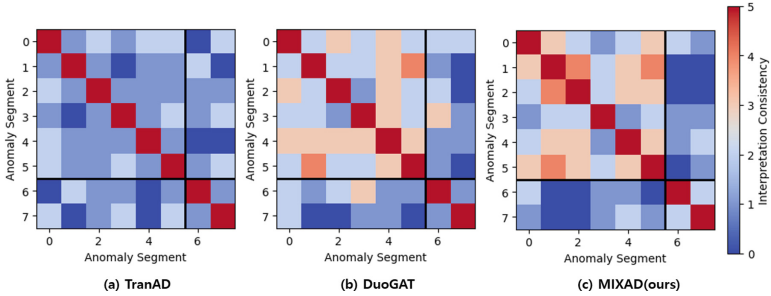
**Fig. 4.** Visualization of memory activation and anomaly scores for an anomaly segment in the SMD dataset.

This figure consists of four sequential graphs. The top graph illustrates the original time series in blue alongside its reconstruction in orange. The second graph shows the memory activation maps (attention scores) across timestamps, followed by two graphs that display anomaly scores for causal and non-causal features, respectively. A green horizontal line in the third graph highlights the maximum anomaly score among the non-causal features, and a red shaded area across all graphs marks the anomaly segment’s duration. This visualization reveals a discernible shift in memory attention within the anomalous period despite the close resemblance between the actual and reconstructed time series. Importantly, anomaly scores based on memory attention are significantly elevated only for the causal features within this segment. This clear differentiation supports the effectiveness of our anomaly scoring approach in accurately identifying and interpreting anomalies, emphasizing its potential utility.

Furthermore, we analyze anomaly score correlations within a specific anomalous segment in the SMD dataset, as shown in Fig. 3. We calculate the Pearson correlation coefficient for anomaly scores  $s'_{seg}$  across nodes over the duration of an anomaly segment  $seg$ , generating an  $N \times N$  correlation matrix. This matrix is visually depicted as a heatmap, where the top-left box, divided by horizontal and vertical black lines, illustrates the correlation among the root cause features. Notably, features responsible for the anomaly demonstrate a high correlation in their scores, underlining the similar memory activation shift each of these causal features exhibits. Thus, this correlation coefficient serves as a basis for facilitating more accurate interpretations.

**Table 3.** Anomaly detection performance evaluation on the Exathlon dataset.

Model	AD1(F1-score)	AD2(F1-score)
<b>TranAD</b>	0.2166	0.2166
<b>DuoGAT</b>	<b>0.9900</b>	0.1296
<b>MIXAD</b>	0.9665	<b>0.1526</b>



**Fig. 5.** Anomaly interpretation performance evaluation on the Exathlon dataset.

## 5 Case Study: Exathlon Dataset and Testbed

To evaluate our model’s efficacy on a real-world benchmark, we used the Exathlon dataset and testbed from [10], comparing MIXAD with the top baselines DuoGAT and TranAD. Exathlon, unlike the small benchmarks used in Sect. 4, has 2,283 dimensions and includes noisy training data for a realistic scenario. We experimented with data from Spark streaming application 1, containing two types of anomalies, each with 6 and 2 segments, respectively. The testbed in [10] supports range-based evaluation, which is not directly applicable to point-based evaluation. Thus, we used a point-based version of AD1 and AD2 metrics from [10]. AD1 is equivalent to our point-adjusted evaluation, while AD2 is non-point-adjusted. Table 3 shows that TranAD fails completely, flagging all timestamps as anomalous. DuoGAT and MIXAD detect anomalies with high f1-scores over 0.95 in AD1, but their performance drops in AD2, indicating reliance on point-adjustment. MIXAD, designed to detect anomaly segments using memory activation shifts, still achieves a higher f1-score in AD2, proving its superiority. Furthermore, MIXAD’s explanations include which features caused anomalies, while Exathlon lacks ground truth feature labels. Therefore, we modified the consistency metric from [10]. Explanations for the same anomaly type should be similar, while those for different types should differ. We extracted the top-5 causal features for each detected anomaly and quantified consistency by counting intersections between segment pairs. In Fig. 5, the six anomaly segments in the upper left are type 1, and the two in the lower right are type 2. Figure 5 shows that MIXAD provides the most consistent explanations for the same type, proving its superiority in interpreting anomalies in the Exathlon dataset.

## 6 Conclusion

In this paper, we introduce *MIXAD*, an interpretable MTS anomaly detection model designed to effectively capture and store node-level prototypes of fine-grained spatiotemporal patterns. Leveraging the STRGC framework and a memory augmentation method, *MIXAD* offers a more accountable approach to understanding complex data relationships. Our novel anomaly scoring technique, which utilizes memory activation pattern analysis, significantly improves interpretability in MTS anomaly detection. While *MIXAD* may not achieve SOTA results in anomaly detection, its interpretability and the empirical insights it provides mark important advancements for future research in the field.

**Acknowledgments.** This work was partly supported by Institute for Information & communication Technology Planning & evaluation (IITP) grants funded by the Korean government MSIT: (RS-2022-II221199, RS-2024-00337703, RS-2022-II220688, RS-2019-II190421, RS-2023-00230337, RS-2024-00356293, RS-2022-II221045, and RS-2021-II212068).




## References

1. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: a survey. *arXiv preprint arXiv:1901.03407* (2019)
2. Chen, X., Deng, L., Zhao, Y., Zheng, K.: Adversarial autoencoder for unsupervised time series anomaly detection and interpretation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 267–275 (2023)
3. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I., et al.: STL: a seasonal-trend decomposition. *J. Off. Stat* **6**(1), 3–73 (1990)
4. Cook, A.A., Misirli, G., Fan, Z.: Anomaly detection for IoT time-series data: a survey. *IEEE Internet Things J.* **7**(7), 6481–6494 (2019)
5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **29** (2016)
6. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4027–4035 (2021)
7. Gong, D., et al.: Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF International Conference On Computer Vision, pp.1705–1714 (2019)
8. He, X., Zhao, K., Chu, X.: AutoML: a survey of the state-of-the-art. *Knowl.-Based Syst.* **212**, 106622 (2021)
9. Hutchison, J., Pham, D.-S., Soh, S.-T., Ling, H.-C.: Explainable network intrusion detection using external memory models. In: Australasian Joint Conference on Artificial Intelligence, pp. 220–233. Springer (2022)
10. Jacob, V., Song, F., Stiegler, A., Rad, B., Diao, Y., Tatbul, N.: Exathlon: a benchmark for explainable anomaly detection over time series. *arXiv preprint arXiv:2010.05073* (2020)

11. Jiang, R., et al.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol.37, pp. 8078–8086 (2023)
12. Kieu, T., Yang, B., Guo, C., Jensen, C.S.: Outlier detection for time series with recurrent autoencoder ensembles. In: IJCAI, pp. 2725–2732 (2019)
13. Kim, Y.G., Yun, J.-H., Han, S., Kim, H.C., Woo, S.S.: Revitalizing self-organizing map: anomaly detection using forecasting error patterns. In: IFIP International Conference on ICT Systems Security and Privacy Protection, pp. 382–397. Springer (2021)
14. Lee, J., Park, B., Chae, D.-K.: DuoGAT: dual time-oriented graph attention networks for accurate, efficient and explainable anomaly detection on time-series. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 1188–1197 (2023)
15. Li, Z., et al.: Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp. 3220–3230 (2021)
16. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint [arXiv:1607.00148](https://arxiv.org/abs/1607.00148)* (2016)
17. Nedelkoski, S., Bogatinovski, J., Mandapati, A.K., Becker, S., Cardoso, J., Kao, O.: Multi-source distributed system data for AI-powered analytics. In: Service-Oriented and Cloud Computing: 8th IFIP WG 2.14 European Conference, ESOC 2020, Heraklion, Crete, Greece, September 28–30, 2020, Proceedings 8, pp. 161–176. Springer (2020)
18. Park, H., Noh, J., Ham, B.: Learning memory-guided normality for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14372–14381 (2020)
19. Sorensen, H.V., Jones, D., Heideman, M., Burrus, C.: Real-valued fast fourier transform algorithms. *IEEE Trans. Acoust. Speech Signal Process.* **35**(6), 849–863 (1987)
20. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 2828–2837 (2019)
21. Tuli, S., Casale, G., Jennings, N.R.: TranAD: deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint [arXiv:2201.07284](https://arxiv.org/abs/2201.07284)* (2022)
22. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
23. Xu, H., et al.: Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In: Proceedings of the 2018 world wide web conference, pp.187–196 (2018)
24. Yu, H., et al.: Regularized graph structure learning with semantic knowledge for multi-variate time-series forecasting. *arXiv preprint [arXiv:2210.06126](https://arxiv.org/abs/2210.06126)* (2022)
25. Zhang, Q., Chang, J., Meng, G., Xiang, S., Pan, C.: Spatio-temporal graph structure learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 1177–1185 (2020)
26. Zhao, H., et al.: Multivariate time-series anomaly detection via graph attention network. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 841–850. IEEE (2020)



# Rough Set Theoretic Approach for Solving the Multi-Armed Bandit Problems

Avinash Paidi<sup>1</sup>(✉) , Istapriya Jagravi<sup>2</sup> , and Pabitra Mitra<sup>3</sup> 

<sup>1</sup> Centre of Excellence in Artificial Intelligence, IIT Kharagpur, Kharagpur, India  
avinashpaidi@gmail.com

<sup>2</sup> Department of Mechanical Engineering, IIT Kharagpur, Kharagpur, India

<sup>3</sup> Computer Science and Engineering, IIT Kharagpur, Kharagpur, India  
pabitra@cse.iitkgp.ac.in

**Abstract.** We propose a Rough set-theoretic approach for solving the stochastic Multi-Armed Bandit (MAB) problems. The proposed approach is a modification to the Epsilon-greedy ( $\epsilon$ -greedy) algorithm used to solve the stochastic multi-armed bandit problems. In our proposed approach, initially, we randomly explore all the arms for some time steps to gather basic reward data for each arm. Using this collected basic reward data, rough estimates of the expected rewards of the arms are calculated. Based on the rough estimates of the expected rewards of all the arms, we partition the arms into three parts following the principles of rough set theory. In the subsequent time steps, different exploration rates are used for different partitions to guide arm selection, to balance between exploring new options and exploiting known performers. We periodically update each arm's estimated mean reward and re-partition them into three parts following a defined process. We continuously monitor for stability in the reward structure of the problem and adaptively adjust the exploration-exploitation balance in response. As the algorithm progresses, the arms with the potential to become the best arm are identified and the exploration is narrowed, which leads to a concentration of effort on arms that consistently yield higher rewards, leaving out the other arms. This strategic selection of arms directs the exploration toward the most promising arms, enhancing the efficiency of the learning process, which is proved with the support of the experimental results.

**Keywords:** Multi-Armed bandits · Rough set ·  $\epsilon$ -greedy algorithm

## 1 Introduction

Multi-armed bandits (MAB) are a class of sequential decision-making problems where an agent must repeatedly choose among multiple actions, often called

---

A. Paidi and I. Jagravi—These two authors contributed equally.

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

A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15309, pp. 258–275, 2025.

[https://doi.org/10.1007/978-3-031-78189-6\\_17](https://doi.org/10.1007/978-3-031-78189-6_17)

arms, to maximize cumulative reward over time. MAB problems find applications in various domains, including online advertising, recommendation systems, clinical trials, resource allocation, and dynamic pricing [1–5]. In a stochastic multi-armed bandit problem, the rewards associated with each arm are not fixed but instead follow a probability distribution. The true value of an arbitrary arm  $a$ , denoted  $q_*(a)$  is the expected reward given that arm  $a$  is selected. If  $A_t = a$  is the arm selected at time step  $t$ , and  $R_t$  is the reward received, the true value of arm  $a$  is:

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a]. \quad (1)$$

One simple approach for calculating the estimates of values of each arm is to calculate the sample average of rewards obtained from each arm. The value estimate  $Q(a)$  for an arm  $a$  can be calculated as the average of the rewards received from selecting arm  $a$  up to time step  $t$ . If  $R_i(a)$  is the reward obtained from selecting arm  $a$  at time step  $i$ ,  $N(a)$  is the number of times the arm  $a$  has been selected up to time step  $t$ , the estimate is calculated as:

$$Q(a) = \frac{\sum_{i=1}^t R_i(a)}{N(a)}. \quad (2)$$

The key challenge in MAB problems is the exploration-exploitation trade-off. The agent must balance between exploring arms to gather information about their reward distributions and exploiting arms that are believed to yield high rewards based on the available data. The algorithms for solving stochastic multi-armed bandit problems vary significantly in how they balance the exploration and exploitation trade-offs.

The follow-the-leader strategy is a simple approach in which the agent continuously selects the arm that has yielded the highest average reward in the past. This strategy focuses purely on exploitation and does not incorporate any exploration mechanism. In the  $\epsilon$ -greedy algorithm, the agent chooses the arm with the highest estimated value most of the time (exploitation), but occasionally (with probability  $\epsilon$ ) selects a random arm to explore. Upper Confidence Bound (UCB) algorithms [6–8] select arms based on an upper confidence bound on their expected rewards. Arms that have higher uncertainty or potential for high rewards are prioritized, balancing exploration and exploitation. In Thompson sampling [9], the agent maintains a probability distribution over the true reward distribution of each arm. At each step, it samples from these distributions and selects the arm with the highest sampled value. This approach naturally balances exploration and exploitation as it inherently captures uncertainty. EXP3 algorithm [10], which is a variation of the exponentially weighted average algorithm, assigns weights to arms based on their past performance and explores arms with lower weights more frequently while exploiting arms with higher weights. The Softmax action selection technique also known as the Boltzmann exploration technique [11] selects actions probabilistically based on their estimated values. Actions with higher estimated values have higher probabilities of being chosen,

but all actions have a chance of being explored. The temperature parameter controls the level of exploration. The Successive rejects algorithm [12] divides the arms into multiple stages and progressively eliminates underperforming arms. Initially, all arms are given some exploration, but as the stages progress, the algorithm exploits the best-performing arms more, thus balancing exploration and exploitation.

In this paper, we introduced a rough set theory-based approach that modifies the  $\epsilon$ -greedy algorithm for solving the stochastic multi-armed bandit problems. The proposed approach balances the exploration-exploitation trade-off by partitioning the arms and using different exploration rates for each partition, and then after making progress and obtaining a stable reward structure, focusing only on a set of arms eliminating the remaining arms from consideration. The ability to identify the set of arms that certainly does not have the potential to become the best arm and eliminate them from further consideration once the stable reward structure is obtained is what sets apart our method from other methods. Eliminating a set of under-performing arms and focusing only on the remaining arms results in better performance and acceleration of the learning process of our proposed method compared to other methods.

## 2 Applying Rough Set Concepts to Stochastic Multi-Armed Bandits

Introduced by Zdzislaw Pawlak, rough set theory [13] is a mathematical framework for dealing with uncertainty and vagueness in data. Rough set theory deals with the approximation of concepts using lower and upper approximations. Lower approximation represents the set of objects that certainly belong to a given concept, while upper approximation represents the set of objects that possibly belong to the concept. The difference between the upper and lower approximations (boundary region) captures the uncertainty or vagueness in the concept. The set of objects that are neither in the lower approximation of a concept nor in the upper approximation of a concept are said to be in the outside region of a concept and they represent the objects that certainly do not belong to the concept. This can be mathematically described as follows. Let  $D = (O, A \cup d)$  be a decision system, where,  $O$  is a set of objects,  $A$  is a set of attributes, and  $d$  is the decision attribute where  $d \notin A$ . If  $F$  is a subset of  $A$ ,  $E$  is a concept which is a subset of  $O$ , we can approximate  $E$  using only the information provided by  $F$  by constructing the  $F$ -lower approximation of  $E$  denoted by  $\underline{F}(E)$  and  $F$ -upper approximation of  $E$  denoted by  $\overline{F}(E)$  respectively. Here,  $\underline{F}(E) = \{x \mid [x]_F \subseteq E\}$  and  $\overline{F}(E) = \{x \mid [x]_F \cap E \neq \emptyset\}$ , where  $[x]_F$  denotes an equivalence class of an element  $x \in E$  with respect to  $F$ . The objects in the  $\underline{F}(E)$  are classified as certain members of  $E$  based on the knowledge of  $F$ , while the objects in the  $\overline{F}(E)$  can be only classified as possible members of  $E$  based on knowledge of  $F$ . The set  $R = \overline{F}(E) - \underline{F}(E)$  is called the  $F$ -boundary region of  $E$  that consists of the objects we can not decisively classify into  $E$  based on

the knowledge of  $F$ . The set  $O - \overline{F}(E)$  is called the  $F$ -outside region of  $E$  and consists of objects that certainly are not part of  $E$ .

In stochastic MAB problems, uncertainty typically arises from not knowing the true reward distributions of each arm. Algorithms used to solve the stochastic MAB problems address this uncertainty in different ways. For example, the UCB algorithms address this uncertainty by maintaining confidence bounds. In our proposed approach, the inherent uncertainty of the stochastic multi-armed bandit problems is reinterpreted as a challenge of identifying arms with the definitive potential to be top performers. We tackle this uncertainty by partitioning all the arms in an MAB into the lower approximation region, the boundary region (difference between upper and lower approximations) and the outside regions based on the available estimates of the expected rewards of all the arms. All the arms in the lower approximation region are identified to be the arms that certainly have the potential to become the best arm. The arms in the boundary region are identified to be the arms that may have the potential to become the optimal arm. The arms in the outside region are the arms that certainly have no potential to become the best arm.

Our proposed approach can be formulated as a decision system that can leverage the rough set theory principles and fits into the mathematical notations discussed earlier in this section as follows. We can define the set of arms as a set of objects  $O$ , available estimates of the expected rewards as the singleton set of attribute  $A$ , and decision attribute  $d$  as the attribute that decides whether an arm has the potential to become the best arm or not. We consider the subset of  $A$ , i.e.  $F$  to be same as  $A$  and the subset of  $O$ , i.e. the concept  $E$  as the set of arms that have the potential to become the best arm. Now, for the concept  $E$ , the arms in the  $\underline{F}(E)$  are classified as members of  $E$  with certainty based on knowledge of  $F$ , while the arms in the  $\overline{F}(E)$  can be only classified as possible members of  $E$  based on knowledge of  $F$ . The set  $R = \overline{F}(E) - \underline{F}(E)$  is called the  $F$ -boundary region of  $E$  that consists of the arms we can not decisively classify into  $E$  based on the knowledge of  $F$ . The set  $O - \overline{F}(E)$  is called the  $F$ - outside region of  $E$  and consists of all the arms that certainly are not part of  $E$ . The detailed procedure of our proposed method is explained in the next section.

### 3 Proposed Methodology

Our proposed method is a modification to the traditional epsilon-greedy approach, tailored for complex multi-armed bandit problems where arms represent different strategies or choices with associated rewards. Detailed Process Flow of the proposed method is as follows:

**Initial Uniform Exploration:** Initially, the algorithm engages in a methodical exploration process, where each arm is randomly pulled until every arm has been selected a specific number of times. This specific number of pulls per arm, determined by the problem's context and user preference, is essential for forming a foundational understanding of each arm's reward distribution. This approach



establishes a baseline performance metric by ensuring a balanced exploration across all arms, allowing the algorithm to accumulate preliminary data that reflects the reward potential of each arm. We should ensure that the timesteps( $t$ ) taken for initial exploration should be much less than total timesteps( $T$ ) i.e.  $t \ll T$ .

**Performance-Based Partitioning of Arms:** Upon completion of the preliminary exploration time steps, arms are partitioned into three parts based on their performance, quantified by their estimated mean rewards which we got after initial exploration. These partitions denote lower approximation arms, boundary region arms, and outside region arms. Lower approximation arms consist of the top 20% arms. These are the high-performing arms, identified as having the maximum or near-maximum reward rates compared to others. They represent the most promising options for exploitation. The boundary region arms consist of the middle 20% arms. These arms exhibit moderate performance. They are neither the best nor the poorest performers, offering a balance between risk and reward. The outside region arms consist of the bottom 60% arms. Constituting the majority, these arms have the lowest performance based on the current estimates. They are less likely to yield high rewards but are crucial for exploration in a changing environment. The top-performing arm which is in the lower approximate region is monitored throughout the time steps. Whenever the top-performing arm changes, this partitioning will be redone based on available estimates of expected rewards of arms to obtain a new set of lower approximation, boundary and outside region arms.

**Adaptive Exploration with Different Rates:** After partitioning the arms into three parts, different exploration rates are applied for each partition using three exploration rates denoted by  $E_0$ ,  $E_1$ , and  $E_2$ .  $E_0$  is the largest, applied to the top 20% arms, reflecting a higher tendency towards exploiting these high-performing arms.  $E_1$  is the next largest, applied to the middle 20% arms, set to encourage exploration among these potentially improving arms, balanced with exploitation.  $E_2$  is the lowest, applied to the bottom 60% arms, encouraging some exploration among these least performing arms to remain responsive to possible shifts in their reward possibilities. This partitioning and the application of exploration rates  $E_0$ ,  $E_1$ , and  $E_2$  to the respective categories are revisited continually in response to shifts in the performance of the arms until a stable reward structure is achieved, ensuring dynamic adaptability and responsiveness throughout the process.

**Dynamic Exploration-Exploitation Shift:** The proposed technique monitors the leading arm within the top 20% segment. If this arm remains the best choice for a predefined number of steps (user-defined and task-dependent), it suggests a stable reward landscape where the current best choice consistently outperforms others. In response to this stability, the algorithm shifts its focus.

The exploration of the bottom 60% arms is halted, while exploration within the top 40% is continued with a more focused approach. In periods of perceived stability, the algorithm intensifies its exploitation of the top-performing arms, thereby capitalizing on the identified reward potential. This shift is crucial in maximizing the cumulative rewards over time, particularly in scenarios where some arms consistently outperform others. Once the stable reward structure is attained, all the arms in the outside region are eliminated from consideration in the future steps. Whereas the best-performing arm in the lower approximate region is pulled with a probability of  $1 - \epsilon$  and with a small probability of  $\epsilon$ , all other arms in the lower approximation regions and boundary region are explored with equal probability. In a way, this is similar to applying the traditional  $\epsilon$ -greedy algorithm on a selected set of arms, eliminating the outside region arms from consideration.

## 4 Experimental Results

To validate the effectiveness of our proposed methodology, we experimented with three distinct problems. These problems, namely the Bandit problem, advertising problem, and election campaign problem, were carefully selected to span a spectrum of scenarios that a multi-armed bandit problem could encapsulate. Each presents a unique challenge, providing a rich ground for testing and proving the robustness of our approach against the traditional methods: Epsilon-Greedy, Upper Confidence Bound (UCB), EXP3, and Boltzmann exploration. The Advertising and Election Campaign datasets are available at <https://github.com/ista24/MAB>.

To analyze the performance of our proposed approach, we focus on three key metrics that collectively offer a comprehensive view of its efficacy. The first is Cumulative Average Rewards, a metric that captures the mean of the rewards accumulated over time, reflecting the long-term benefit of the strategy employed. The second metric, % Optimal Action, provides insight into the algorithm's precision in selecting the best possible action given the circumstances. It is a direct measure of the algorithm's learning capabilities and its ability to make the most informed decisions over the course of its operation. Lastly, the Cumulative Regret metric sums up the opportunity loss over time, considering what could have been earned had the best decision been made at every time step. In our assessment, we focus on the Cumulative Regret based on the expected reward. By relying on expected rewards, we ensure that our evaluation is not subject to the volatility that can arise from the random nature of actual rewards, which may not accurately reflect the strategic efficacy of the algorithm. These metrics together construct a narrative that allows us to critically assess and compare the proposed methodology's performance, not just in isolation but also in direct competition with traditional  $\epsilon$ -greedy, UCB, EXP3, and Boltzmann exploration methods.

In our proposed method, the initial exploration of arms is done till the time step where every arm is pulled at least 5 times and E0, E1 and E2 values are set

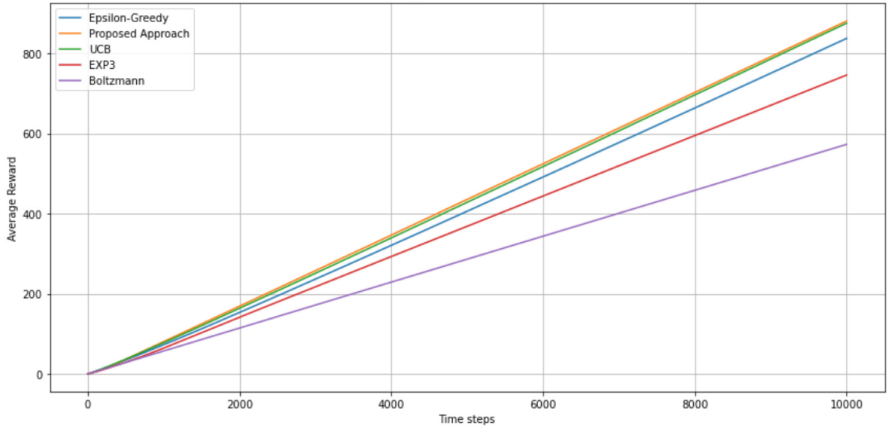
to 0.5, 0.4, and 0.1 respectively for all the experiments. Also, it is assumed that a stable reward structure is obtained if the top-performing arm is not changed for 200 consecutive time steps for all the experiments. The  $\epsilon$  values for our proposed method and the  $\epsilon$ -greedy algorithm are set to decay linearly starting from an initial value of 0.1 to a minimum value of 0.001. The learning rate of the EXP3 method is set to  $\eta = \sqrt{\frac{\ln K}{KT}}$ , where  $K$  is the number of arms and  $T$  is the total number of time steps and the temperature parameter in the Boltzmann exploration is set to 1. All the plots are generated after averaging the results obtained by running the algorithms for 2000 runs on the problems. Each run consists of 10000 time steps in all the experiments.

#### 4.1 Bandit Problem

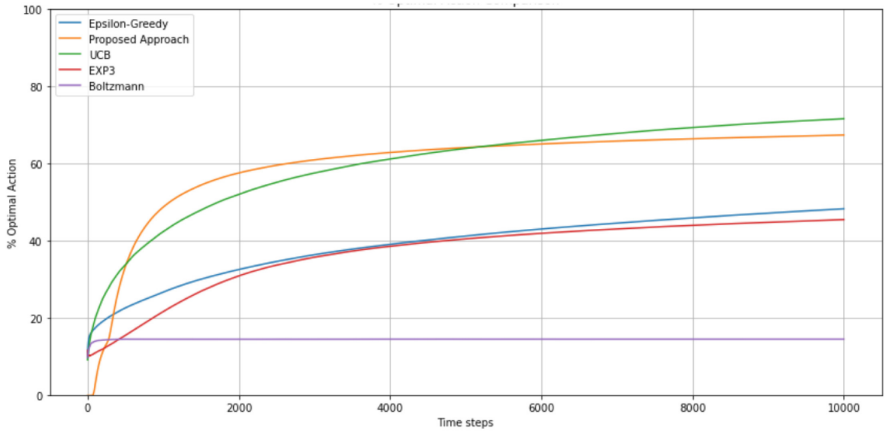
The Bandit problem comprises a set of slot machines, metaphorically known as “bandits,” each with a lever that, when pulled, yields a reward drawn from a probability distribution unique to that bandit. In our case, we have constructed a bandit scenario with 10 arms, each representing a unique bandit. For each arm, the expected reward is selected in accordance with normal distribution with mean 0 and unit variance, introducing a balanced level of uncertainty across all arms. The primary challenge here is to pinpoint the arm that consistently yields the highest rewards over numerous trials, thereby maximizing gains.

Our Proposed Approach is adept at navigating this problem, characterized by its uncertainty and the need for strategic decision-making. We evaluate its effectiveness through three critical performance metrics: Cumulative Average Rewards, Percentage of Optimal Actions, and Cumulative Regret.

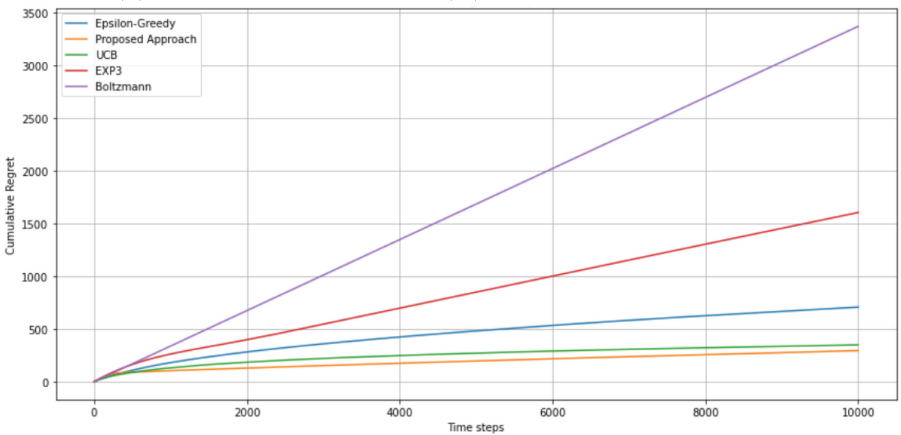
From Fig. 1, it can be seen that the proposed approach demonstrated a consistently higher average reward compared to other strategies. This improvement suggests that the modified selection method, which dynamically adjusts its exploration and exploitation balance, effectively identifies and leverages the more rewarding arms in the bandit problem. In terms of strategic acumen, the % Optimal Action Comparison graph reveals that the Proposed Approach swiftly learns to identify and exploit the most rewarding options, outpacing the other algorithms significantly. This rapid ascension and sustained high percentage of optimal actions underscore a more profound understanding and quicker adaptation to the environment’s reward structure. Perhaps most notably, the Cumulative Regret Comparison depicts a more nuanced success. While all algorithms exhibit an inevitable increase in regret over time due to the exploratory nature of the task, our approach incurs the lowest regret, implying that it is the most efficient at minimizing potential losses. The comparatively lower trajectory of the Proposed Approach’s regret line serves as evidence of its ability to leverage past experiences to make increasingly more informed and lucrative decisions as time progresses.



(a) Comparison of Cumulative Average Rewards



(b) Comparison of percentage (%) of times optimal action taken



(c) Comparison of Cumulative Regrets

**Fig. 1.** Performance comparison of proposed method with  $\epsilon$ -greedy, UCB, EXP3, and Boltzmann exploration algorithms on bandit problem.

## 4.2 Advertising Problem

The Advertising problem simulates an advertising scenario where various campaigns, represented as arms in the multi-armed bandit framework, are evaluated based on historical data. In this framework, there are 10 distinct campaigns(arms), each with its own click-through rate(CTR) which serves as a measure of its effectiveness. These CTRs are determined by aggregating past user interactions-total clicks and displays for each advertisement campaign. Upon initializing the environment, the system processes the dataset, calculating the CTR for each ad by dividing the total number of clicks it received by the total number of times it was displayed. The ad with the highest CTR is flagged as the 'optimal' ad since it has demonstrated the greatest likelihood of engaging users.

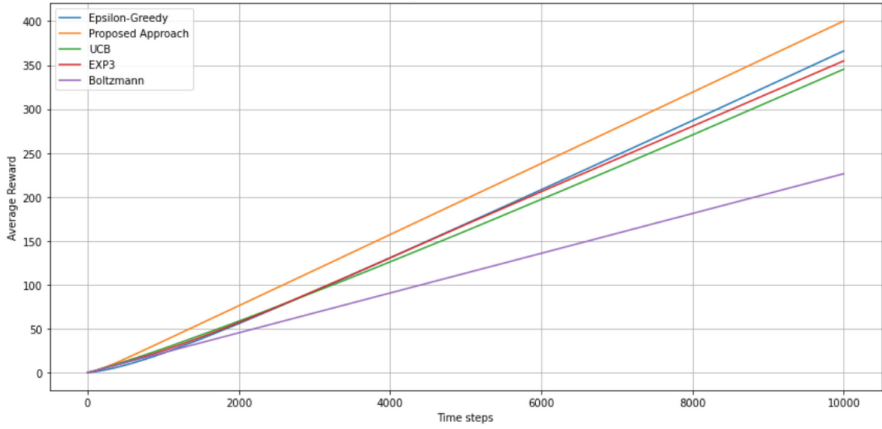
When the model simulates choosing an ad to display, which corresponds to pulling an arm in bandit terminology, it mimics the real-life binary outcome of whether a displayed ad will be clicked. This is modelled as a Bernoulli trial, where the probability of success (a click) is equal to the ad's historical CTR. Such a setup mirrors the uncertain nature of user response in actual advertising campaigns.

In this dynamic problem, the main objective is to identify which ad, amongst other choices, will continue to perform best in terms of user engagement. This task is inherently challenging due to the variability in CTRs and the probabilistic nature of user clicks. To assess the effectiveness of different strategies within this environment, several performance metrics are employed. Cumulative Average rewards measure the success rate across all ad displays, while the percentage of optimal actions gauges how often the best-performing ad is chosen. Cumulative regret, focused solely on the expected rewards, captures the potential loss incurred from not always selecting the optimal ad over time. These metrics allow for a nuanced evaluation of an algorithm's performance, taking into account both immediate outcomes and long-term strategic learning.

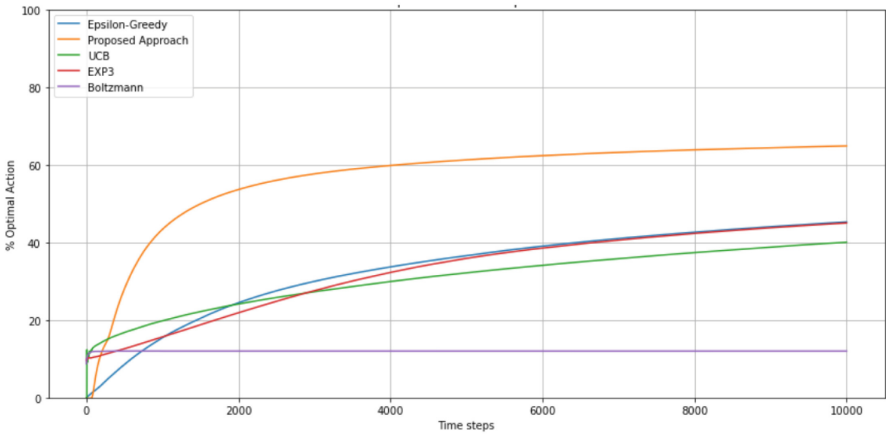
Upon examining the results of our simulations across the Advertising problem, it is apparent that the proposed approach consistently outshines other strategies. The graphs in Fig. 2 clearly illustrate a robust trend of superior performance by the Proposed Approach.

## 4.3 Election Campaign Problem

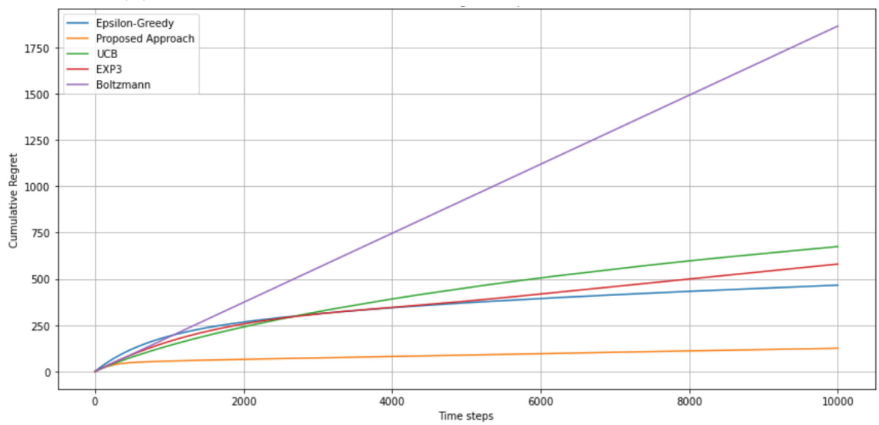
The Election Campaign problem simulates the strategic decision-making process of an election campaign, where different outreach efforts, each represented as one of 24 distinct "arms", are evaluated for their effectiveness. This environment is derived from a dataset that includes the historical performance of these 24 campaign strategies. Each strategy is represented as an "arm," and the effectiveness of these strategies is quantified by their associated rewards-akin to the level of voter engagement or the number of votes garnered. Upon initializing the environment, the rewards are normalized against the highest recorded reward to create a set of probabilities. These probabilities reflect the expected success rate of each campaign strategy when employed. The challenge in this problem lies in



(a) Comparison of Cumulative Average Rewards



(b) Comparison of percentage of times optimal actions taken



(c) Comparison of Cumulative Regrets

**Fig. 2.** Performance comparison of proposed method with  $\epsilon$ -greedy, UCB, EXP3, and Boltzmann exploration algorithms on Advertising problem.

determining which strategy or arm is likely to yield the most significant engagement or the highest number of votes, a task that is inherently probabilistic due to the unpredictable nature of voter behaviour. The simulation of reward outcomes is based on these probabilities, using a Bernoulli distribution to model the binary success of a campaign effort: a '1' indicating a successful outcome (effective engagement or vote acquisition) and a '0' indicating an unsuccessful one. The problem's key objective is to navigate through the probabilistic landscape of campaign strategies to find the most effective approach for voter engagement.

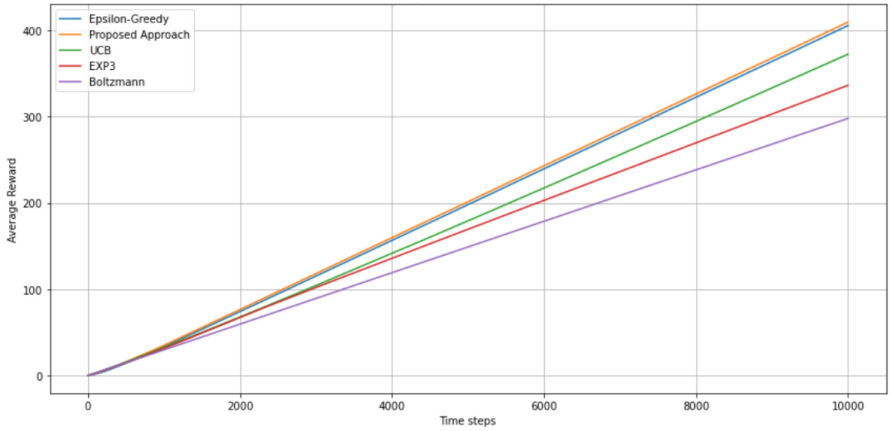
From the plots in Fig. 3, we can observe that the proposed approach demonstrates a significant improvement over the other methods across all performance metrics in the Election Campaign problem.

In conclusion, the comprehensive analysis conducted across three distinct problems conclusively demonstrates the superior performance of our proposed approach over the traditional Epsilon-Greedy, UCB, EXP3, and Boltzmann exploration methods. This consistent outperformance is evident in all the measured metrics - Cumulative Average Reward, Percentage of Optimal Action, and Cumulative Regret based on expected rewards-across Bandit, Advertising, and Election Campaign problems.

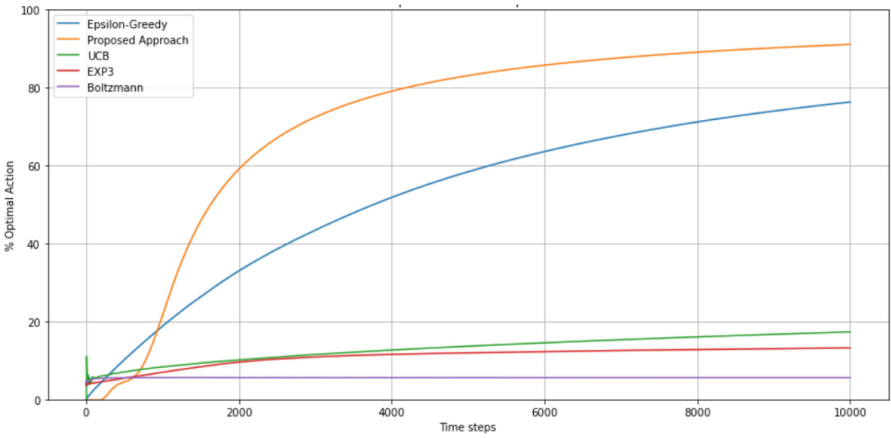
#### 4.4 Ablation Study

We conducted ablation studies to understand the impact of various components on the performance of our proposed method for solving stochastic multi-armed bandit problems. We have compared the performance of our baseline model with four variant models on all three environments. All 4 variants generate different initial estimates of the expected rewards of arms used for dividing the arms based on rough set principles. Variant 1 and Variant 2 are exclusively designed to understand the impact of the initial exploration phase in the proposed method. Variant 3 and Variant 4 are designed to understand the impact of partitioning of arms in different ways. In variant 3, the arms are partitioned into the top 15 percentile arms, the next 15 percentile arms and the remaining 70 percentile arms. Whereas in Variant 4, the arms are partitioned into the top 25 percentile arms, the next 25 percentile arms and the remaining 50 percentile arms. In all 4 variants, all the other parameters and experimental setups are kept the same as in our proposed method baseline model used in the comparison with other methods earlier in this section. The resulting plots are shown in Figs. 4, 5, and 6.

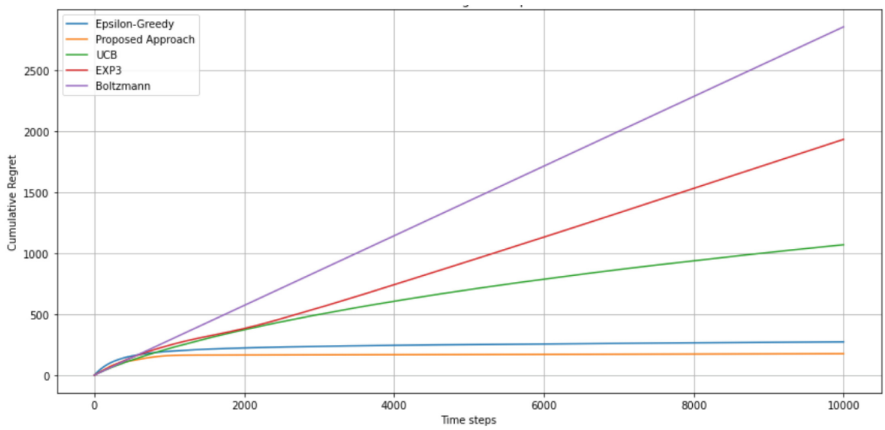
From the ablation studies, it can be observed that the different initial estimates of expected rewards of arms used for partitioning the arms have no major impact on the performance of the proposed algorithm. The ablation of different proportions of arms resulted in a moderate impact, highlighting its role in accurate reward estimation and effective exploitation and suggesting its utility might be more context-specific or synergistic with other components.



(a) Comparison of Cumulative Average Rewards



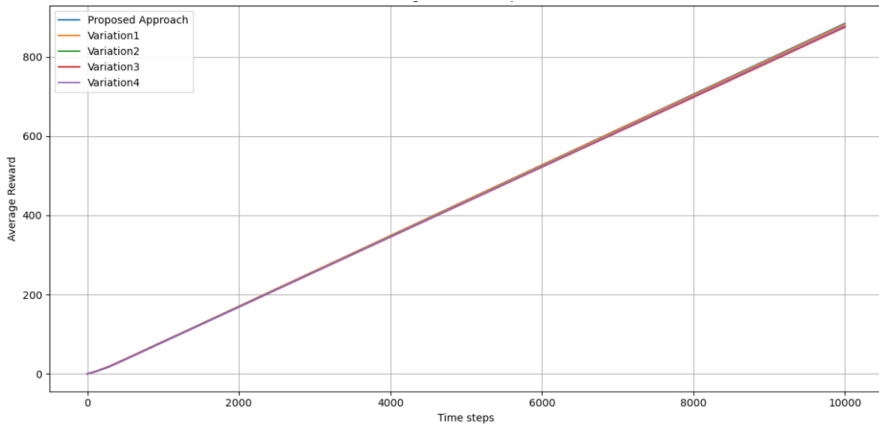
(b) Comparison of percentage of optimal actions taken



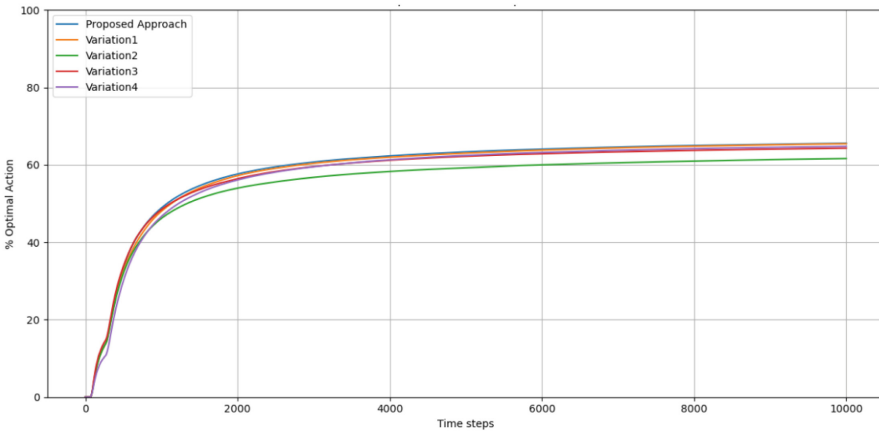
(c) Comparison of Cumulative Regrets

**Fig. 3.** Performance comparison of proposed method with  $\epsilon$ -greedy, UCB, EXP3, and Boltzmann exploration algorithms on Election campaign problem.

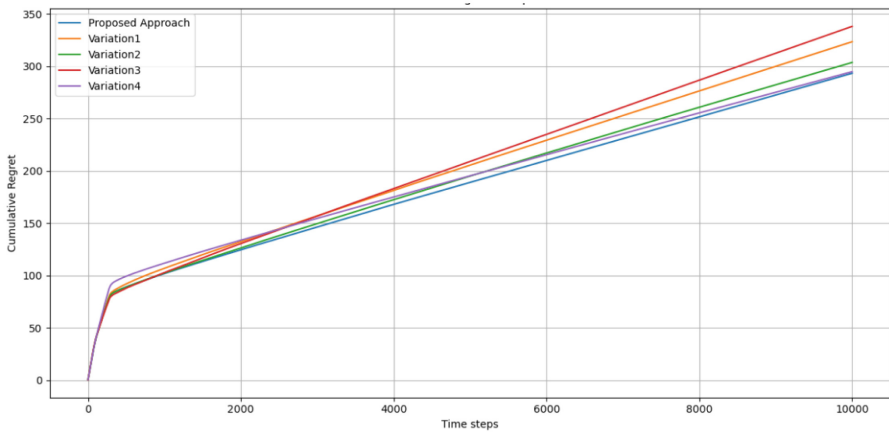




(a) Comparison of Cumulative Average Rewards

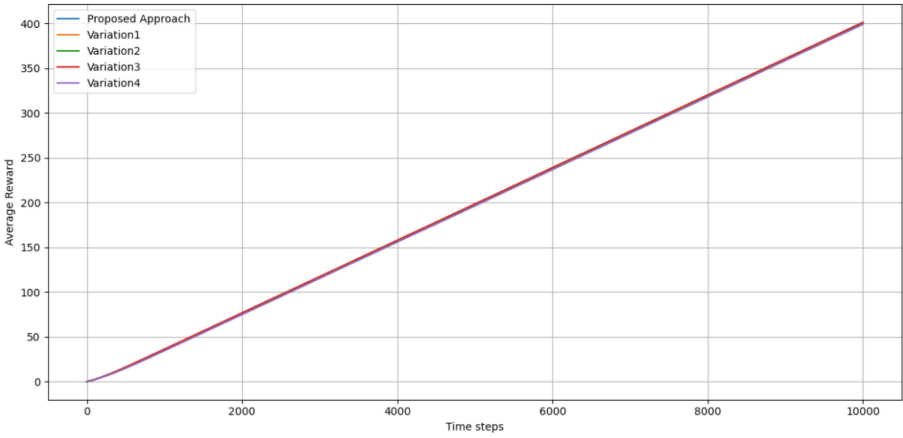


(b) Comparison of percentage of optimal actions taken

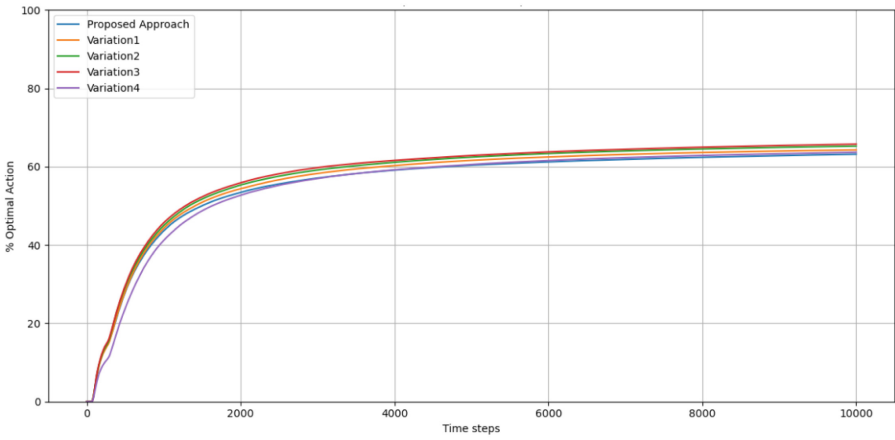


(c) Comparison of Cumulative Regrets

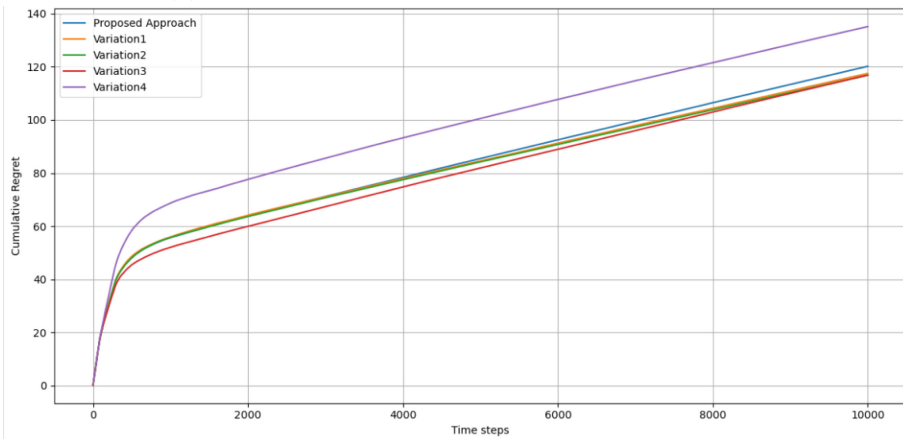
**Fig. 4.** Ablation study on the bandit problem.



(a) Comparison of Cumulative Average Rewards

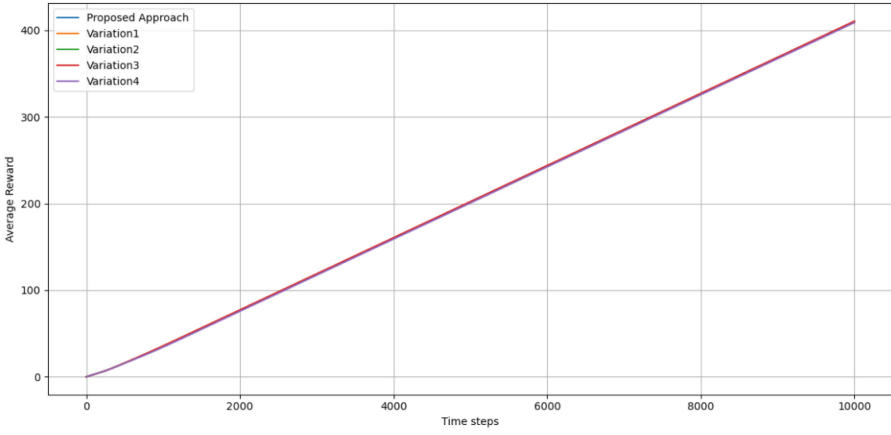


(b) Comparison of percentage of optimal actions taken

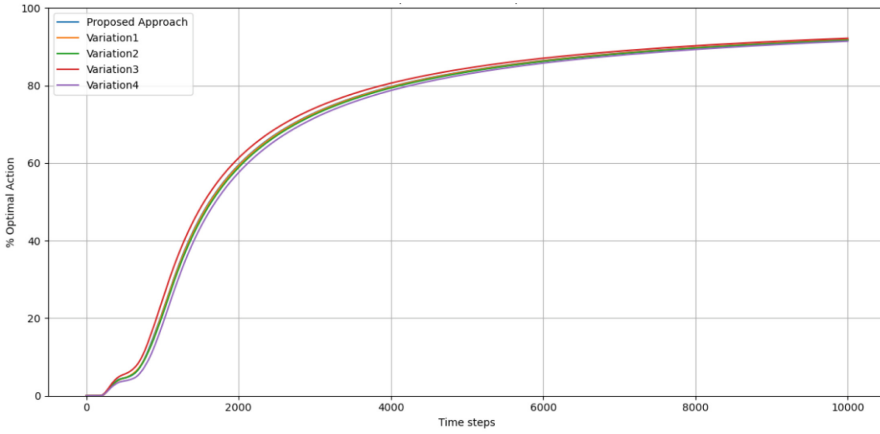


(c) Comparison of Cumulative Regrets

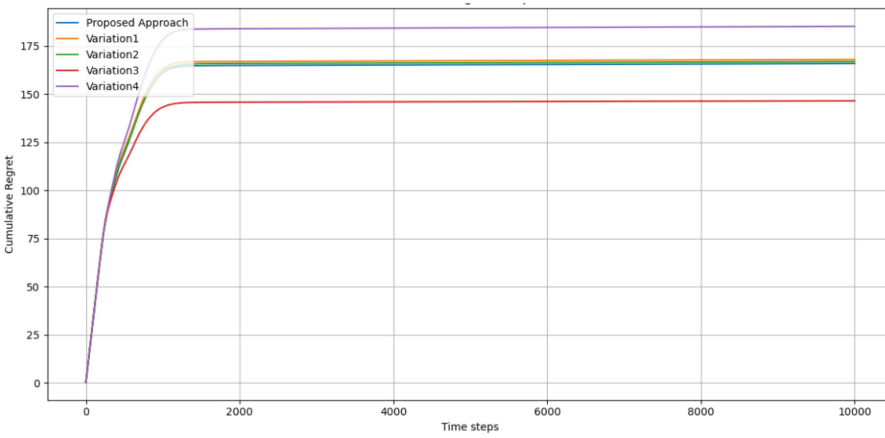
**Fig. 5.** Ablation study on the advertising problem.



(a) Comparison of Cumulative Average Rewards



(b) Comparison of percentage of optimal actions taken



(c) Comparison of Cumulative Regrets

**Fig. 6.** Ablation study on the Election campaign problem.

## 4.5 Discussions

**Adaptive Partitioning for Diverse Situational Requirements:** Our proposed approach, designed to initially partition arms in a 20-20-60 proportion, also empowers the users with the flexibility to adapt to scenarios with fewer arms or skewed reward distribution of arms. This adaptability is crucial, as it aligns with the core principles of rough set theory, which forms the basis for our partitioning strategy, allowing for a more nuanced and situationally aware application of the method.

In situations where the total number of arms is less than five, adhering to a predefined 20-20-60 partitioning structure becomes infeasible. Our proposed method accommodates this by offering the user the flexibility to adjust the partitioning proportion based on the actual number of arms available. The users can reallocate the arms into lower approximation, boundary region, and outside region categories in a manner that truly represents their performance hierarchy. This user-led adjustment ensures that each arm is categorized in a way that accurately reflects its performance potential, even in scenarios with a smaller number of arms. By doing so, the method retains its strategic effectiveness, ensuring that decision-making is attuned to the specific context of the environment, no matter how few the choices may be.

In scenarios where reward distributions are skewed - for instance, when a single arm vastly outperforms the others, or when the top-performing arms have marginally different rewards - rigidly sticking to the standard 20-20-60 partitioning rule may not effectively capture the nuances of the situation. In such cases, our approach empowers users to redefine the partitioning proportions dynamically. The users can reallocate the arms into lower approximation, boundary region, and outside region categories in a manner that truly represents their performance hierarchy in this case as well. This flexibility in partitioning allows for a more precise and adaptive strategy, ensuring that the algorithm's choices are always congruent with the prevailing reward dynamics of the arms.

By enabling this level of user-guided adaptability, our method not only stays true to the tenets of rough set theory but also broadens its applicability to a wide array of multi-armed bandit scenarios, thereby establishing itself as a robust and adaptable solution for diverse real-world challenges.

**Impact of the Initial Uniform Exploration Phase on the Method's Performance:** From the ablation study presented, it is observed that irrespective of the different initial estimates of the expected rewards, the proposed method finds the optimal arm in an efficient manner. It can be concluded that even if the initial exploration phase is not optimally executed, the algorithm is still capable of identifying the optimal arm at later time steps. But this might come at the cost of delay in obtaining a stable reward structure to proceed with eliminating the arms that certainly do not have the potential to become the best arm. This proves the reliability and robustness of the proposed approach.

## 5 Conclusions and Future Work

In this paper, we have presented a modified  $\epsilon$ -greedy technique based on the rough set-theoretic approach. We have leveraged the principles of rough set theory to handle uncertainty in the multi-armed bandit problems. The modified  $\epsilon$ -greedy technique identified the optimal arm much earlier and yielded a high cumulative average reward and less cumulative regret than not just the  $\epsilon$ -greedy algorithm, but also the UCB, EXP3, and Boltzmann exploration algorithms.

One important research direction involves making other multi-armed bandit algorithms like Thompson sampling, Bayesian UCB, EXP3 and Boltzmann exploration efficient by incorporating the rough set theory principles into them. Another interesting research direction involves using the rough set theory principles to assist in feature selection, dimensionality reduction, and decision-making based on contextual information in contextual multi-armed bandits.

## References

1. Schwartz, E.M., Bradlow, E.T., Fader, P.S.: Customer acquisition via display advertising using multi-armed bandit experiments. *Mark. Sci.* **36**(4), 500–522 (2017)
2. Zeng, C., Wang, Q., Mokhtari, S., Li, T.: Online context-aware recommendation with time-varying multi-armed bandit. In: *Proceedings of the 22nd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, pp. 2025–2034 (2016)
3. Villar, S.S., Bowden, J., Wason, J.: Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Stat. sci. rev. j. Inst. Math. Stat.* **30**(2), 199 (2015)
4. Youssef, M.J., Veeravalli, V.V., Farah, J., Nour, C.A., Douillard, C.: Resource allocation in NOMA-based self-organizing networks using stochastic multi-armed bandits. *IEEE Trans. Commun.* **69**(9), 6003–60 (2021)
5. Misra, K., Schwartz, E.M., Abernethy, J.: Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Mark. Science* **38**(2), 226–252 (2019)
6. Carpentier, A., Alessandro L., Ghavamzadeh, M., Munos, R., Auer, P.: Upper-confidence-bound algorithms for active learning in multi-armed bandits. In: *International Conference on Algorithmic Learning Theory*, pp. 189–203. Berlin, Heidelberg: Springer Berlin Heidelberg (2011)
7. Audibert, J.Y., Munos, R., Szepesvári, C.: Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.* **410**(19), 1876–1902 (2009)
8. Jamieson, K., Malloy, M., Nowak, R., Bubeck, S.: *lil'ucb*: an optimal exploration algorithm for multi-armed bandits. In: *Conference on Learning Theory*, pp. 423–439. PMLR (2014)
9. Agrawal, S., Goyal, N.: Analysis of Thompson sampling for the multi-armed bandit problem. In: *Conference on Learning Theory*, pp. 39–1. JMLR Workshop and Conference Proceedings (2012)
10. Zhou, D., Tomlin, C.: Budget-constrained multi-armed bandits with multiple plays. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1. (2018)

11. Tokic, M., Palm, G.: Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In: Annual Conference on Artificial Intelligence, pp. 335–346. Berlin, Heidelberg: Springer Berlin Heidelberg, (2011)
12. Audibert, J.-Y., Bubeck, S.: Best arm identification in multi-armed bandits. In: COLT-23th Conference on Learning theory-2010, pp. 13–p (2010)
13. Pawlak, Z.: Rough sets: Theoretical Aspects of Reasoning About Data. Vol. 9. Springer Science and Business Media (2012)



# Hybrid Graph Representation Learning: Integrating Euclidean and Hyperbolic Space

Lening Li, Lei Luo, and Yanguang Sun<sup>(✉)</sup>

Nanjing University of Science and Technology, Nanjing, China  
Lilening@njjust.edu.cn, syg1513@163.com

**Abstract.** Graph representation learning aims to capture the structural and relational information in graphs. Recently, Euclidean space-based methods have achieved tremendous success. However, Euclidean space exhibits structural distortion problems when modeling graph data with tree structures or hierarchy, limiting the model's performance. Based on this, researchers introduce Hyperbolic space to preserve the original structural information, but computations in Hyperbolic space rely on inverse trigonometric functions, resulting in increased computational complexity. How to learn in multi-spaces and make use of their advantages deserves careful consideration. This paper proposes a Hybrid Graph Representation Learning (HGRL) model that trains in the Hyperbolic and Euclidean spaces jointly. Euclidean space possesses the ability to learn regular geometric and has efficient computations, while Hyperbolic spaces are better suited for representing hierarchical and non-linear relationships. Technically, we utilize the Euclidean contrast loss to minimize distances between similar samples, helping tight clusters in the traditional space. Simultaneously, the Hyperbolic hierarchy loss and Hyperbolic uniformity loss enable the model to comprehend intricate hierarchical relationships and ensure a uniform distribution of the data on the Poincaré Ball. Extensive experiments in node classification, clustering, and visualization tasks demonstrate the effectiveness of the HGRL mode in capturing hierarchy structures. We also employed ablation studies to validate the indispensability of each component.

**Keywords:** Representation learning · Euclidean space · Hyperbolic space

## 1 Introduction

Graph data consists of tree-like and hierarchical structures, which is a universal language for representing and embedding complex systems. With its unique properties, graph data has been widely adopted in many real-world tasks, such as recommendation systems [25, 40, 47], knowledge graph embeddings [3, 14, 26, 41], single-cell RNA sequence analysis [17, 37, 44], and so on. However, raw graph data

is difficult to apply directly in the field of machine learning due to the redundant information or the inherent complexity [43]. Learning how to properly encode the variety of graph data can help people understand the world better.

Graph representation learning can effectively simplify high-dimensional raw graph data into low-dimensional dense vectors [15], and retain the structural and semantic features of the graph data. A large number of graph representation learning algorithms [13, 16, 23, 35, 39] have been proposed to generate excellent embeddings. Early, some graph representation learning algorithms [12, 20, 49] learn embeddings from Euclidean space, which has the advantage of high computational efficiency [45]. For example, GCC [6] offers a variant that maintains a computational complexity akin to k-means, while matching the efficiency of the GCN [16] propagation matrix, which makes the GCN a low-pass filter.

However, the performance is often unsatisfactory when embedding datasets with non-Euclidean geometric characteristics, because Euclidean space has zero curvature, which fails to model the hierarchical information. The researchers attempted to utilize the power of hyperbolic representations to solve the above problem [5, 19, 30, 46, 51]. Hyperbolic space is a non-Euclidean space that contains constant negative curvature. It has the property of exponential expansion and the hierarchical, tree-like structure [11]. The number of leaf nodes increases exponentially with depth, similar to the exponential growth of the hyperbolic surface area with radius. In comparison, growth in Euclidean space is polynomial [21]. So hyperbolic-based representation learning is emerging as a compelling field of study, garnering increasing interest and attention from the research community.

Despite Hyperbolic space has great performance for tree-like graph data, it is restricted by complex computation. We hope to combine the advantages of Euclidean space to learn a better representation of graph structure [31, 52]. According to the above analysis, embedding diverse and complex real-world graph data in multiple spaces is a promising direction in representation learning. There are already some attempts [9, 18, 21, 22, 45], but these representation learning methods encounter significant challenges, *e.g.*, inefficiency or the complex algorithm with low interpretability. GIL [52] leverages both hyperbolic and Euclidean topological features and derives a novel distance-aware propagation and interaction learning scheme. But GIL provides a limited explanation for its prediction. Therefore, we need to further investigate how to effectively utilize multiple spaces to ensure the rational exploitation of various geometric properties.

To address the mentioned limitations above and to make multi-space representation learning more reliable, in this paper, we propose a hybrid graph representation learning (HGRL) model that acts as a link between Hyperbolic and Euclidean spaces. By learning the geometric property of both spaces, the model can explore complex graph structures and relationships in the real world, leading to improved accuracy and better generalization. Specifically, we first use a hyperbolic encoder to extract the low-dimensional features from the input data. Secondly, we introduce the Euclidean contrastive loss using the adjacency matrix to maximize the distance between different classes and minimize the dis-



tance between the same classes in Euclidean space. Finally, considering the hierarchical information present in Hyperbolic space, we construct two Hyperbolic losses to preserve both the hierarchical and uniform structures of the datasets. Our network architecture is more straightforward than previous methods. Significantly, we utilize two novel Hyperbolic losses and an optimized Euclidean contrast loss to preserve diverse information about the dataset effectively. In the following sections, we concretely describe the role and principles of each loss, providing a comprehensive explanation of the HGRL model. We also demonstrate the advantages of our model by many downstream tasks and ablation study.

To sum up, the contributions of this work can be summarized as follows:

We propose a hybrid graph representation learning (HGRL) model that exploits the embedding of graph data in both Euclidean and Hyperbolic spaces. It leverages the advantages of both spaces for performance improvement.

We construct the hierarchy loss and uniformity loss functions in Hyperbolic space for perceiving the faithful hierarchical structure and ensuring uniform distribution in Hyperbolic space to preserve more information of the data.

We conduct extensive experiments to demonstrate the effectiveness of the proposed HGRL model on citation datasets and biological datasets. We design rich ablation experiments to demonstrate the role and necessity of each part.

## 2 Related Work

### 2.1 Representation Learning

Nowadays, representation learning receives significant attention. We divide it into matrix factorization methods and deep learning methods [15]. For the former, existing methods usually rely on factorizing the feature matrix to learn embeddings. For example, COLES [51] extended the Laplacian Eigenmaps with contrastive learning and minimizes a surrogate of Wasserstein distance. For the latter type of method, GCN [16] minimized the reconstruction error for training the encoder. DIM [13] formulates graph neural networks that focus on Mutual Information (MI) and optimize through the contrast of local and global graph features to acquire the graph feature representations. Later, DGI [39] maximized the MI between patch representations and high-dimension graph summary.

However, none of these approaches investigate the inherent hierarchical information of the data, which we believe could enhance the training performance. To solve this issue, we introduce the Hyperbolic space to learn representations.

### 2.2 Hyperbolic Representation Learning

Researchers have shown significant interest in hierarchical representation learning using non-Euclidean geometry. LaBNE [1] generalized eigenvalues by using the Laplacian spectral matrix. Angular coalescence [28] defined graph weights based on local topological information, determines angular coordinates of nodes using methods like IsoMAP [36], LLE [33] ultimately obtained the graph representation in Hyperbolic space. HNN [8] extended the Euclidean-based network

layer into the Hyperbolic space. GCN [4] also utilized the GCN with hyperbolic geometry to learn embeddings. It implements the operator in Euclidean space to hyperbolic models and employs trainable curvatures at each layer to transform input features into embeddings in Hyperbolic space.

These methods extend the Euclidean-based approach to Hyperbolic space. Unlike that, we retain the learning in Euclidean space and design more suitable loss objectives to supplement the shortcomings of Euclidean space based on the fact that hierarchies are more readily learned in Hyperbolic space.

### 3 Preliminaries

#### 3.1 Poincaré Ball Model

The Poincaré disk model is a two-dimensional model of hyperbolic geometry, encompassing all points on a unit disk. A common generalization is the Poincaré ball model [7]. Due to the geometric properties of the Poincaré ball model, it can effectively learn entities hierarchical structures and similarity relationships of the data. We choose the Poincaré ball model as our basic model. The model with  $d$  dimensions and constant negative curvature  $c$  is formally called the Riemannian manifold  $\mathbb{H}_c^d = (\mathcal{H}_c^d, \mathfrak{g}_p^c)$ , where  $\mathcal{H}_c^d = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| < 1\}$  represents the open ball and  $\mathfrak{g}_p^c$  is a metric tensor.  $\|\cdot\|$  is the Euclidean norm. For  $\mathbf{x} \in \mathbb{H}_c^d$ , the relation between the Riemannian metric tensor  $\mathfrak{g}_p^c$  and the Euclidean metric tensor  $\mathfrak{g}_e$  is defined as follows:

$$\mathfrak{g}_p^c(\mathbf{x}) = \left( \frac{2}{1 - c\|\mathbf{x}\|^2} \right)^2 \mathfrak{g}_e(\mathbf{x}), \quad \mathfrak{g}_e(\mathbf{x}) = \frac{(1 - c\|\mathbf{x}\|^2)^2}{4} \mathfrak{g}_p^c(\mathbf{x}). \quad (1)$$

In the Poincaré ball model, we compute the hyperbolic distance between two points by measuring the arc. Specifically, for two points  $\mathbf{x}, \mathbf{y} \in \mathbb{H}_c^d$ , the hyperbolic distance between them is formally defined as follows:

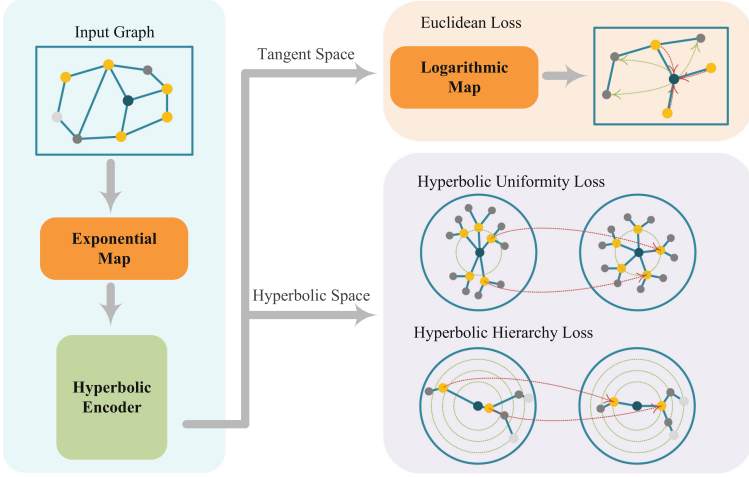
$$d(\mathbf{x}, \mathbf{y}) = \operatorname{arcosh} \left( 1 + 2 \frac{\|\mathbf{x} - \mathbf{y}\|^2}{(1 - \|\mathbf{x}\|^2)(1 - \|\mathbf{y}\|^2)} \right). \quad (2)$$

#### 3.2 Gyrovector Spaces

The vector spaces serve as the algebraic foundation for Euclidean geometry, facilitating straightforward vector computations like addition, subtraction, and scalar multiplication. Similarly, the Gyrovector spaces [38] allows the graceful non-associative algebraic forms for hyperbolic geometry which play the same role as vector spaces in the Euclidean geometry [8].

For  $\mathbf{x}, \mathbf{y} \in \mathbb{H}_c^d$ , the Mbius addition is defined as follows:

$$\mathbf{x} \oplus_c \mathbf{y} := \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}, \quad (3)$$



**Fig. 1.** The overview of the HGRL. It consists of encoder and loss divided into three components: Euclidean contrast loss, Hyperbolic hierarchy loss, and Hyperbolic uniformity loss. During training, we minimize the Euclidean contrast loss to ensure the Euclidean distance between similar samples is as small as possible. Simultaneously, the Hyperbolic hierarchy loss facilitates the learning of hierarchical structures of the data. Additionally, by introducing Hyperbolic uniformity loss, our objective is to achieve a uniform distribution of data on the Poincaré Ball.

where  $c$  is the curvature of the Hyperbolic space, and when  $c = 0$  the Eq. 3 degenerates to the addition in the Euclidean space.

For  $c > 0$ , the Mbius scalar multiplication of  $\mathbf{x} \in \mathbb{H}_c^d$  by  $r \in \mathbb{R}$  is defined as follows:

$$r \otimes_c \mathbf{x} := (1/\sqrt{c}) \tanh(r \tanh^{-1}(\sqrt{c}\|\mathbf{x}\|)) \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad (4)$$

where  $r \otimes_c \mathbf{0} := \mathbf{0}$ . Similar to the Mbius addition, this Eq. 4 converges to the Euclidean scalar multiplication when  $c \rightarrow 0$ .

We utilize the hyperbolic encoder to learn the embeddings in the Hyperbolic latent space. Then we leverage the Euclidean characteristics of the embeddings. The tangent space  $\mathcal{T}_x \mathbb{H}_c^d$  is a vector space with the same dimensional as  $\mathbb{H}_c^d$ . The mapping  $\mathcal{T}_x \mathbb{H}_c^d \rightarrow \mathbb{H}_c^d$  is the exponential map, while the inverse  $\mathbb{H}_c^d \rightarrow \mathcal{T}_x \mathbb{H}_c^d$  is known as the logarithmic map. We chose the origin as the target point because the mapping function at the origin exhibits symmetry. Given the  $\mathbf{u} \in \mathcal{T}_x \mathbb{H}_c^d$  and the  $\mathbf{v} \in \mathbb{H}_c^d$  and the operations mentioned earlier, the closed-form formulations of mapping can be defined as follows:

$$\mathbf{v} = \exp_0^c(\mathbf{u}) = \tanh(\sqrt{c}\|\mathbf{u}\|) \frac{\mathbf{u}}{\sqrt{c}\|\mathbf{u}\|}, \quad (5)$$

$$\mathbf{u} = \log_0^c(\mathbf{v}) = \arctan(\sqrt{c}\|\mathbf{v}\|) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}. \quad (6)$$

## 4 Method

The overview of the HGRL framework is shown in Fig. 1. We try to optimize performance in multi-space by leveraging different loss components collectively.

### 4.1 Hyperbolic Encoder

Because of the better hierarchy learning ability of the Hyperbolic space and the complete operation in the Gyrovector spaces, we adopt the hyperbolic encoder. It has the same structures as the Euclidean encoder but is conducted by the hyperbolic operation. First, we map the raw datasets into the Hyperbolic space before inputting them into the network. This is because all the raw datasets are in the Euclidean space currently. In particular, given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $n = |\mathcal{V}|$  nodes and  $m = |\mathcal{E}|$  edges and hyperbolic encoder  $M_c(\cdot)$ , the  $\mathcal{G}$  will be embedded as follows:

$$\mathbf{H}^h = M_c(\exp_0^c(\mathcal{G})), \quad (7)$$

$$\mathbf{H}^e = \log_0^c(\mathbf{H}^h). \quad (8)$$

where  $\mathbf{H}^e$  and  $\mathbf{H}^h$  represent the output in Euclidean space and Hyperbolic space respectively, which can be used for downstream tasks.

### 4.2 Euclidean Loss

For the datasets without adjacency matrix  $\mathbf{A}$ , we employ the k-Nearest Neighbor classification algorithm to construct it. For each node, we reserve k nodes as its neighborhood and use distance as the weight of the adjacency matrix. To maximize the distance between different classes and minimize the distance between the same classes, we use the contrastive Laplacian eigenmaps loss in the Euclidean space [51].

$$\mathcal{L}_{le} = \text{Tr}(\mathbf{H}^{e\top} \mathbf{L} \mathbf{H}^e) - \frac{\gamma}{n} \sum_{i=1}^n \text{Tr}(\mathbf{H}^{e\top} \mathbf{L}_i^{(-)} \mathbf{H}^e), \quad (9)$$

where  $\mathbf{L} = \mathbf{I} - \mathbf{A}^{(+)}$  and  $\mathbf{L}_i^{(-)} = \mathbf{I} - \mathbf{A}_i^{(-)}$ ,  $i = 1, \dots, n$  are randomly generated as degree-normalized Laplacian matrices with the negative sampling. The scalar  $\gamma$  controls the impact of negative sampling.

However, in some cases, this loss function may yield negative values, potentially leading to misleading assessments of model performance. To address this issue, we introduce an exponential term into the Euclidean loss function, to ensure that the loss function always produces non-negative values. This allows the model to account for different loss components in a balanced manner throughout the entire training process, consequently strengthening the learning and generalization capabilities of HGRL. The final loss Euclidean defined as follows:

$$\mathcal{L}_{euc} = \exp(\mathcal{L}_{le}). \quad (10)$$

### 4.3 Hyperbolic Hierarchy Loss

While Euclidean loss effectively discerns differences between classes, it does not preserve the overall structures of the data. So we leverage the hyperbolic norm to learn the hierarchical structure of the data in the Hyperbolic latent space. The hyperbolic paradigm is defined as the distance of the vector to the origin which allows it to naturally capture the hierarchical structure of the data in the Poincaré disk model. According to Eq. 2, the hyperbolic norm is defined as follows:

$$\|\mathbf{x}\|_h = \operatorname{arcosh}\left(1 + \frac{2\|\mathbf{x}\|^2}{1 - \|\mathbf{x}\|^2}\right). \quad (11)$$

Our objective is to keep the norm of high-dimensional data and its embedding at a constant level. We minimize the difference between the hyperbolic representation  $\exp_0^c(\mathcal{G})$  of the input graph and the embedding representation  $\mathbf{H}^h$  obtained from the network. So the Hyperbolic hierarchical loss is defined as follows:

$$\mathcal{L}_{hie} = \mathbb{E}(\|\exp_0^c(\mathcal{G})\|_h - \|\mathbf{H}^h\|_h)^2. \quad (12)$$

### 4.4 Hyperbolic Uniformity Loss

The contrastive loss comprises uniformity and alignment losses [42]. To simplify the computations, we employ the inner product rather than distance as a measure of similarity. The reason for this is the conformality property of the Poincaré ball model, *i.e.*, the angles between any embeddings in the Poincaré ball as the same as those in Euclidean space. The hyperbolic inner product is defined as follows:

$$\langle \mathbf{x}, \mathbf{y} \rangle_h = \|\mathbf{x}\|_h \cdot \|\mathbf{y}\|_h \cdot \cos\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\|_h \cdot \|\mathbf{y}\|_h \cdot \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}, \quad (13)$$

where  $\langle \cdot, \cdot \rangle$  denotes the inner product in Euclidean space and  $\|\cdot\|_h$  is the hyperbolic norm as defined by Eq. 11.

According to the above definition, we employ the uniformity loss to ensure the uniform distribution of the dataset over the Poincaré Ball, preserving maximal information about the data. Considering the Gaussian potential kernel, the uniformity loss is defined as follows:

$$\mathcal{L}_{uni} = \log \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p_{\text{data}}} \left[ e^{-\tau \cdot \langle \mathbf{x}, \mathbf{y} \rangle_h} \right], \quad \tau > 0, \quad (14)$$

where  $p_{\text{data}}$  is the distribution of data on the Poincaré ball model and  $\tau$  is a temperature parameter.

### 4.5 Total Loss

In summary, we maximize inter-class distance and minimize intra-class distance through Euclidean loss, learn the data hierarchical structure by using Hyperbolic hierarchy loss, and preserve the most information with uniformity loss. Consequently, the total loss for HGRL is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{euc} + \alpha \mathcal{L}_{hie} + \beta \mathcal{L}_{uni}, \quad (15)$$

where  $\alpha$  and  $\beta$  represent trade-off hyper-parameters to balance the effects of the Euclidean loss and the two Hyperbolic losses in multi-space learning.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** As shown in Table 1, we construct the node classification and clustering tasks on four widely-used datasets to evaluate the performance of our HGRL model: Cora [24], Citeseer [10], Pubmed [34], and Cora Full [2]. All of them are citation networks, which edges are citation links and nodes are documents. We visualize the results on biological datasets: the mouse myelopoiesis scRNA-seq dataset [29] which has been preprocessed into 9 types [32].

**Table 1.** The summary of the dataset.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
Cora Full	19,793	65,311	8,710	70
scRNA-seq	382	-	532	9

**Metrics.** For node classification tasks, we measure the performance by the mean classification accuracy and the standard deviation. We gather the result over 50 random splits and conduct the experiments with different sample sizes per class, practically 5 and 20 samples per class. Toward the node clustering task, we use the clustering Accuracy (Acc), Normalized Mutual Information (NMI), and macro F1-score (F1) over 10 random splits. For all datasets, we use the logistic regression classifier and the KMeans respectively.

**Experimental Settings.** We set the dimension to be the same as COLES [51], *e.g.*, 512 dimensions on Cora, Cora Full and Citeseer, and 256 dimensions on Pubmed. For our model, we set the hyperbolic curvature  $c = 1$ . And since the Poincaré Ball has a Riemannian manifold structure, we optimize Eq. 15 via stochastic Riemannian optimization methods RSGD.

### 5.2 Node Classification and Clustering

Table 2 and Table 3 show the experiments of node classification and node clustering respectively. CO-N and CO-S represent COLES-GCN [51] and COLES-SSGC [51] respectively. Simple Spectral Graph Convolution (SSGC) [50] is a

**Table 2.** The mean classification accuracy (%) and the standard deviation over 50 random splits. For each dataset, the experiments have different sample sizes per class, *i.e.*, 5 and 20 samples per class. The best accuracy result is **red**.

Method	Cora		Citeseer		Pubmed		Cora Full	
	(5)	(20)	(5)	(20)	(5)	(20)	(5)	(20)
GCN [16]	67.5 ± 4.8	79.4 ± 1.6	57.7 ± 4.7	69.4 ± 1.4	65.4 ± 5.2	77.2 ± 2.1	49.3 ± 1.8	61.5 ± 0.5
DGI [39]	72.9 ± 4.0	78.1 ± 1.8	65.7 ± 3.6	71.1 ± 1.1	65.3 ± 5.7	73.9 ± 2.3	50.5 ± 1.4	58.4 ± 0.6
SSGC [50]	71.4 ± 4.4	81.3 ± 1.2	60.3 ± 4.0	69.5 ± 1.2	67.6 ± 4.2	73.3 ± 2.0	41.8 ± 1.7	60.0 ± 0.5
CO-G [51]	73.8 ± 3.4	80.8 ± 1.3	66.0 ± 2.6	69.0 ± 1.3	62.7 ± 4.6	72.7 ± 2.1	47.3 ± 1.5	58.9 ± 0.5
CO-S [51]	76.5 ± 2.6	81.5 ± 1.2	67.5 ± 2.2	71.3 ± 1.0	66.0 ± 5.2	77.4 ± 1.9	50.8 ± 1.4	61.8 ± 0.5
HGCN [4]	76.1 ± 1.6	77.6 ± 0.7	57.6 ± 4.7	64.8 ± 0.8	67.1 ± 7.6	72.0 ± 1.0	52.3 ± 2.3	63.6 ± 0.7
HIE [48]	77.5 ± 2.6	81.6 ± 0.8	67.9 ± 4.1	71.5 ± 0.8	<b>74.9 ± 7.0</b>	76.3 ± 1.0	53.3 ± 2.2	<b>65.1 ± 0.8</b>
HGRL	74.6 ± 3.1	79.1 ± 0.9	67.7 ± 3.0	71.5 ± 0.9	64.1 ± 4.9	73.7 ± 2.6	49.9 ± 1.5	60.1 ± 0.5
HGRL-S	<b>78.7 ± 2.0</b>	<b>83.6 ± 0.9</b>	<b>68.8 ± 2.5</b>	<b>71.7 ± 0.8</b>	67.7 ± 4.6	<b>77.6 ± 1.7</b>	<b>53.3 ± 1.4</b>	61.7 ± 0.6

spectral filter to obtain the local and global information of each node before training. Considering the refined graph structure understanding and feature learning facilitated by SSGC, we also extend our framework with SSGC [50] and named HGRL-S. It can be seen that compared with the method based on a single space, our HGRL method has the best performance in most cases.

From Table 2, the proposed HGRL method has smaller standard deviations indicating that our HGRL is more stable and reliable. Specifically, compared to the method GCN, we achieved an average reduction of 0.95 in the standard deviation. On the Cora and Citeseer, our method outperforms the Hyperbolic-based HIE by 1.2%, 2%, 0.9%, and 0.2% classification accuracy respectively. Our method also outperforms the Euclidean-based COLES by 2.2%, 2.1%, 1.3%, and 0.4% respectively. Despite HIE has the better results on the pubmed and cora full, it introduces instability. Similarly, the proposed HGRL method achieves excellent performance in the clustering task.

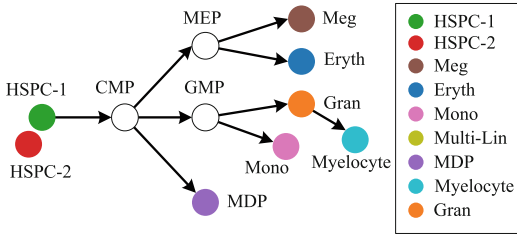
### 5.3 Visualization

Single-cell RNA sequencing (scRNA-seq) is used to analyze gene expression in individual cells, revealing differences between different cells and providing a deeper understanding of cell types and functions. Therefore, it needed to uncover the developmental trajectory of cells along a tree-like structure with multiple branches. We conducted a visualization experiment to demonstrate the performance of dimension reduction ability on the scRNA-seq dataset. Specifically, we reduced the dataset from 512 dimensions to two-dimensional embeddings.

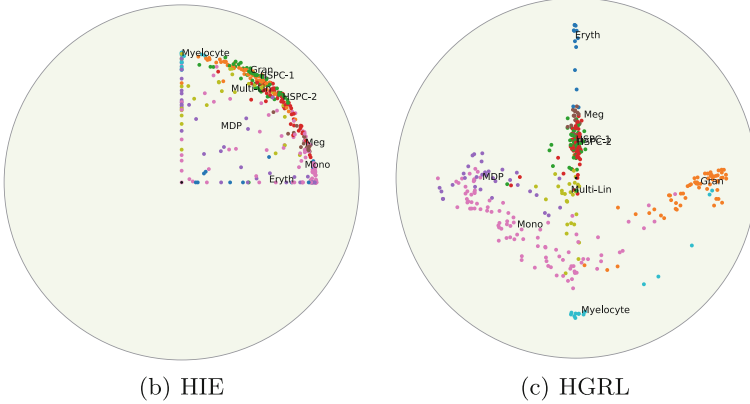
The results are shown in Fig. 2. Figure 2(a) is the canonical hematopoietic cell lineage tree [27]. Figure 2(b) and Fig. 2(c) are the visualizations of the HIE [48] and HGRL respectively.

**Table 3.** The Acc(%), NMI(%), and F1(%) performance of clustering task on Cora, Citeseer, and Pubmed. The best accuracy result is red.

Method	Cora			Citeseer			Pubmed		
	Acc%	NMI%	F1%	Acc%	NMI%	F1%	Acc%	NMI%	F1%
GCN [16]	59.05	43.06	59.38	45.97	20.08	45.57	61.88	25.48	60.70
SSGC [50]	68.96	54.22	65.43	69.11	42.87	64.65	68.18	31.82	67.81
CO-G [51]	60.74	45.49	59.33	63.28	37.54	59.17	63.46	25.73	63.42
CO-S [51]	69.70	55.35	63.06	69.20	44.41	64.70	68.76	33.42	68.12
GCC [6]	74.29	59.17	70.35	69.45	45.13	64.54	70.82	32.30	69.89
HGCN [4]	75.48	54.01	75.41	63.81	38.39	62.45	71.65	33.28	71.08
HIE [48]	75.72	57.91	75.24	68.85	44.83	64.67	<b>73.31</b>	34.31	<b>73.15</b>
HGRL	70.09	47.66	67.08	68.71	43.30	65.14	65.71	34.64	67.58
HGRL-S	<b>76.85</b>	<b>61.21</b>	<b>75.61</b>	<b>69.76</b>	<b>45.25</b>	<b>65.56</b>	70.11	<b>35.10</b>	69.99



(a) Canonical hematopoietic cell lineage tree [27]



(b) HIE

(c) HGRL

**Fig. 2.** The visualization experiments for HIE and HGRL. HGRL preserves the better structure information. (a) The hierarchical structure of the data. (b) The visualization result of the HIE. (c) The visualization result of the proposed HGRL.



## 5.4 Ablation Study

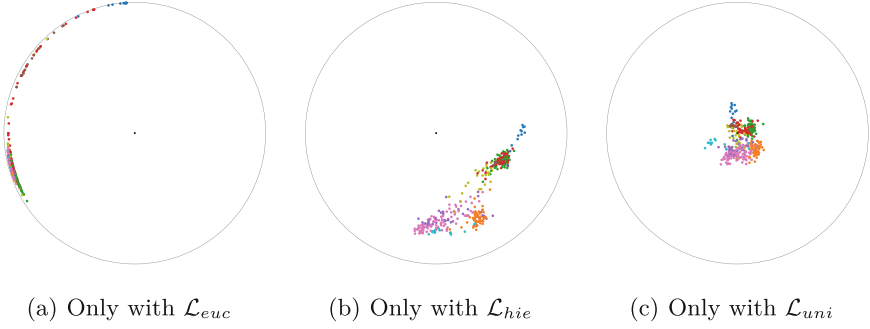
HGRL model considers three types of loss to generate powerful embeddings, *i.e.*, Euclidean contrastive loss, hierarchy loss, and uniformity loss. We investigate the effectiveness of each loss and conduct ablation experiments on the Cora and Citeseer datasets. We consider the role of each loss and the interaction between them. As summarized in Table 4,  $\mathcal{L}_{euc}$  and  $\mathcal{L}_{hie}$  both contribute to the HGRL. The performance decreases when only using the  $\mathcal{L}_{hie}$  because it cannot distinguish different categories. However, it can adjust the distortions caused by other losses and maintain the original structural information of the data.

**Table 4.** The ablation experiment on the Cora and Citeseer dataset.

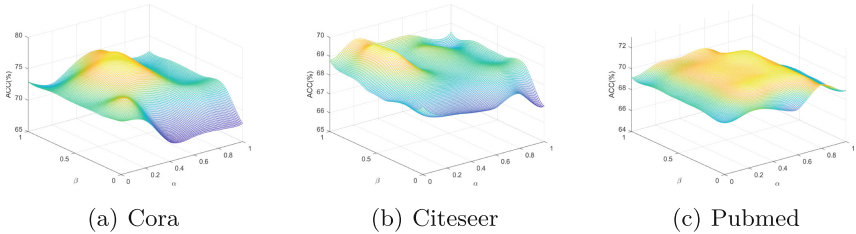
$\mathcal{L}_{euc}$	$\mathcal{L}_{hie}$	$\mathcal{L}_{uni}$	Cora		Citeseer	
			(5)	(20)	(5)	(20)
✓			77.6 ± 3.2	82.2 ± 1.0	67.3 ± 2.3	71.4 ± 1.4
	✓		69.3 ± 4.2	79.1 ± 1.3	60.6 ± 6.2	63.7 ± 2.1
		✓	75.9 ± 2.6	82.3 ± 1.0	61.9 ± 6.5	67.8 ± 1.9
✓	✓	✓	<b>78.71 ± 2.0</b>	<b>83.6 ± 0.9</b>	<b>68.8 ± 2.5</b>	<b>71.7 ± 0.8</b>

As noted, the gap between considering Euclidean contrastive loss alone and the proposed method is relatively small. This is because Cora and Citeseer are citation datasets with sparse connections between nodes. So hierarchical structure is not intuitively clear. To visualize the ability of each part, we conducted visualization ablation experiments on the scRNA-seq dataset using three loss functions individually. Specifically, Fig. 3(a) indicates poor outcomes in the Poincaré disk due to the exclusive application of Euclidean loss. Figure 3(b) using Hyperbolic loss, maintains hierarchical information across classes but lacks clear distinctions and exhibits uneven distribution within the disk. Additionally, Fig. 3(c) achieves a uniform data distribution while preserving maximum information content. Comparing the results in Fig. 2(c), our model effectively balances these losses and successfully utilizes their advantages respectively for representation learning.

**Hyper-Parameter Experiment.** As depicted in the Fig. 4, we set the values of  $\alpha$  and  $\beta$  in Eq. 15 to  $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$  respectively. Our method performs better when setting the larger values for  $\alpha$  and  $\beta$ , and ACC decreases as the  $\alpha$  and  $\beta$  tend to 0. This further demonstrates the effectiveness of our HGRL method.



**Fig. 3.** The visualization ablation experiment on the scRNA-seq dataset.



**Fig. 4.** The clustering ACC(%) for different hyper-parameters ( $\alpha$  and  $\beta$ ) settings.

## 6 Conclusion

As a summary, this paper introduces a hybrid graph representation learning model, named HGRL, which addresses the limitations of existing methods by training in Euclidean and Hyperbolic spaces. We novelly analyze the interaction between three loss functions, *e.g.*, Euclidean contrastive loss, and Hyperbolic hierarchy loss and uniformity loss, achieving a better way to exploit their geometric advantages. The proposed HGRL method demonstrates excellent results in various downstream tasks such as node classification, clustering, and visualization. In particular, it exhibits strong hierarchical learning capabilities through visualization experiments. Our method is an unsupervised learning model, and such models tend to have the issue of unclear boundaries when learning the similarity of nodes. We will continue to study this problem in future work.

**Acknowledgement.** This paper was supported by the National Natural Science Foundation of China (NSFC) under grant No.62276135.

## References

1. Alanis-Lobato, G., Mier, P., Andrade-Navarro, M.A.: Efficient embedding of complex networks to hyperbolic space via their laplacian. *Sci. Rep.* **6**(1), 30108 (2016)
2. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint [arXiv:1707.03815](https://arxiv.org/abs/1707.03815) (2017)
3. Chami, I., Wolf, A., Juan, D.C., Sala, F., Ravi, S., Ré, C.: Low-dimensional hyperbolic knowledge graph embeddings. arXiv preprint [arXiv:2005.00545](https://arxiv.org/abs/2005.00545) (2020)
4. Chami, I., Ying, Z., Ré, C., Leskovec, J.: Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems* **32** (2019)
5. Chen, J., Jin, Z., Wang, Q., Meng, H.: Self-supervised 3d behavior representation learning based on homotopic hyperbolic embedding. *IEEE Trans. Image Process.* **32**, 6061–6074 (2023)
6. Fettal, C., Labiod, L., Nadif, M.: Efficient graph convolution for joint node representation learning and clustering. In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 289–297 (2022)
7. Forti, A.: *Intorno alla vita ed alle opere di Luigi Lagrange discorso letto nel R. Liceo Galilei di Pisa per la festa letteraria commemorativa dal cav. Angelo Forti...* Tipografia delle scienze matematiche e fisiche (1869)
8. Ganea, O., Bécigneul, G., Hofmann, T.: Hyperbolic neural networks. *Advances in neural information processing systems* **31** (2018)
9. Ge, S., Mishra, S., Kornblith, S., Li, C.L., Jacobs, D.: Hyperbolic contrastive learning for visual representations beyond objects. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6840–6849 (2023)
10. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: an automatic citation indexing system. In: *Proceedings of the Third ACM Conference on Digital Libraries*, pp. 89–98 (1998)
11. Hamann, M.: On the tree-likeness of hyperbolic spaces. In: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 164, pp. 345–361. Cambridge University Press (2018)
12. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. arXiv preprint [arXiv:1709.05584](https://arxiv.org/abs/1709.05584) (2017)
13. Hjelm, R.D., et al.: Learning deep representations by mutual information estimation and maximization. arXiv preprint [arXiv:1808.06670](https://arxiv.org/abs/1808.06670) (2018)
14. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. *Adv. Neural. Inf. Process. Syst.* **33**, 22118–22133 (2020)
15. Ju, W., et al.: A comprehensive survey on deep graph representation learning. *Neural Networks*, p. 106207 (2024)
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
17. Klimovskaia, A., Lopez-Paz, D., Bottou, L., Nickel, M.: Poincaré maps for analyzing complex hierarchies in single-cell data. *Nat. Commun.* **11**(1), 2966 (2020)
18. Kovács, B., Palla, G.: Model-independent embedding of directed networks into euclidean and hyperbolic spaces. *Commun. Phys.* **6**(1), 28 (2023)
19. Li, H., Jiang, H., Ye, D., Wang, Q., Du, L., Zeng, Y., Wang, Y., Chen, C., et al.: Dhgat: hyperbolic representation learning on dynamic graphs via attention networks. *Neurocomputing* **568**, 127038 (2024)
20. Li, M.M., Huang, K., Zitnik, M.: Graph representation learning in biomedicine and healthcare. *Nature Biomed. Eng.* **6**(12), 1353–1369 (2022)

21. Lin, F., Bai, B., Guo, Y., Chen, H., Ren, Y., Xu, Z.: Mhcn: a hyperbolic neural network model for multi-view hierarchical clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16525–16535 (2023)
22. Liu, R., Zhang, J., Gao, G.: Multi-space channel representation learning for mono-to-binaural conversion based audio deepfake detection. *Information Fusion* **105**, 102257 (2024)
23. Lu, C., Reddy, C.K., Ning, Y.: Self-supervised graph learning with hyperbolic embedding for temporal health event prediction. *IEEE Trans. Cybern.* **53**(4), 2124–2136 (2021)
24. McCallum, A.K., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. *Inf. Retrieval* **3**, 127–163 (2000)
25. Mirvakhabova, L., Frolov, E., Khrulkov, V., Oseledets, I., Tuzhilin, A.: Performance of hyperbolic geometry models on top-n recommendation tasks. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 527–532 (2020)
26. Montella, S., Rojas-Barahona, L., Heinecke, J.: Hyperbolic temporal knowledge graph embeddings with relational and time curvatures. arXiv preprint [arXiv:2106.04311](https://arxiv.org/abs/2106.04311) (2021)
27. Murphy, K., Weaver, C.: *Janeway’s immunobiology*. Garland science (2016)
28. Muscoloni, A., Thomas, J.M., Ciucci, S., Bianconi, G., Cannistraci, C.V.: Machine learning meets complex networks via coalescent embedding in the hyperbolic space. *Nat. Commun.* **8**(1), 1615 (2017)
29. Olsson, A., et al.: Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. *Nature* **537**(7622), 698–702 (2016)
30. Park, J., Cho, J., Chang, H.J., Choi, J.Y.: Unsupervised hyperbolic representation learning via message passing auto-encoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5516–5526 (2021)
31. Pei, H., Wei, B., Chang, K.C.C., Lei, Y., Yang, B.: Geom-gcn: geometric graph convolutional networks. arXiv preprint [arXiv:2002.05287](https://arxiv.org/abs/2002.05287) (2020)
32. Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., Trapnell, C.: Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods* **14**(10), 979–982 (2017)
33. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
34. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Mag.* **29**(3), 93–93 (2008)
35. Song, J., Park, J., Yang, E.: Tam: topology-aware margin loss for class-imbalanced node classification. In: International Conference on Machine Learning pp. 20369–20383. PMLR (2022)
36. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
37. Tian, T., Zhong, C., Lin, X., Wei, Z., Hakonarson, H.: Complex hierarchical structures in single-cell genomics data unveiled by deep hyperbolic manifold learning. *Genome Res.* **33**(2), 232–246 (2023)
38. Ungar, A.A.: Gyrovector spaces and their differential geometry. *Nonlinear Funct. Anal. Appl* **10**(5), 791–834 (2005)
39. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint [arXiv:1809.10341](https://arxiv.org/abs/1809.10341) (2018)
40. Wang, L., Hu, F., Wu, S., Wang, L.: Fully hyperbolic graph convolution network for recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 3483–3487 (2021)

41. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
42. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: *International Conference on Machine Learning*, pp. 9929–9939. PMLR (2020)
43. Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., Liu, H.: Graph learning: a survey. *IEEE Trans. Artif. Intell.* **2**(2), 109–127 (2021)
44. Xu, Y., Zang, Z., Xia, J., Tan, C., Geng, Y., Li, S.Z.: Structure-preserving visualization for single-cell rna-seq profiles using deep manifold transformation with batch-correction. *Commun. Biol.* **6**(1), 369 (2023)
45. Yang, H., Chen, H., Pan, S., Li, L., Yu, P.S., Xu, G.: Dual space graph contrastive learning. In: *Proceedings of the ACM Web Conference 2022*, pp. 1238–1247 (2022)
46. Yang, M., Zhou, M., Li, Z., Liu, J., Pan, L., Xiong, H., King, I.: Hyperbolic graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852* (2022)
47. Yang, M., Zhou, M., Liu, J., Lian, D., King, I.: Hrcf: enhancing collaborative filtering via hyperbolic geometric regularization. In: *Proceedings of the ACM Web Conference 2022*, pp. 2462–2471 (2022)
48. Yang, M., Zhou, M., Ying, R., Chen, Y., King, I.: Hyperbolic representation learning: Revisiting and advancing. In: *Proceedings of the 40th International Conference on Machine Learning, ICML'23* (2023)
49. Yang, Y., Zuo, X., Das, A., Xu, H., Zheng, W.: Representation learning of biological concepts: a systematic review. *Curr. Bioinform.* **19**(1), 61–72 (2024)
50. Zhu, H., Koniusz, P.: Simple spectral graph convolution. In: *International Conference on Learning Representations* (2020)
51. Zhu, H., Sun, K., Koniusz, P.: Contrastive laplacian eigenmaps. *Adv. Neural. Inf. Process. Syst.* **34**, 5682–5695 (2021)
52. Zhu, S., Pan, S., Zhou, C., Wu, J., Cao, Y., Wang, B.: Graph geometry interaction learning. *Adv. Neural. Inf. Process. Syst.* **33**, 7548–7558 (2020)



# Learning Object Focused Attention

Vivek Trivedy<sup>(✉)</sup>, Amani Almalki, and Longin Jan Latecki

Temple University, Philadelphia, PA 19122, USA  
{vivek.trivedy,amani.almalki,latecki}@temple.edu

**Abstract.** We propose an adaptation to the training of Vision Transformers (ViTs) that allows for an explicit modeling of objects during the attention computation. This is achieved by adding a new branch to selected attention layers that computes an auxiliary loss which we call the object-focused attention (OFA) loss. We restrict the attention to image patches that belong to the same object class, which allows ViTs to gain a better understanding of configural (or holistic) object shapes by focusing on intra-object patches instead of other patches such as those in the background. Our proposed inductive bias fits easily into the attention framework of transformers since it only adds an auxiliary loss over selected attention layers. Furthermore, our approach has no additional overhead during inference. We also experiment with multiscale masking to further improve the performance of our OFA model and give a path forward for self-supervised learning with our method. Our experimental results demonstrate that ViTs with OFA achieve better classification results than their base models, exhibit a stronger generalization ability to out-of-distribution (OOD) and adversarially corrupted images, and learn representations based on object shapes rather than spurious correlations via general textures. For our OOD setting, we generate a novel dataset using the COCO dataset and Stable Diffusion inpainting which we plan to share with the community.

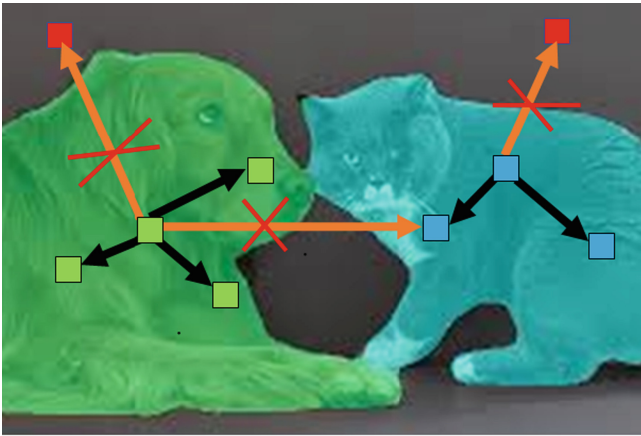
**Keywords:** representation learning · vision transformers · attention mechanism

## 1 Introduction

One of the key ideas of vision transformers (ViTs) is to update the representation of a given patch  $p$  as a weighted sum of feature vectors from all image patches. The weights, which are computed using the transformer attention mechanism, are determined based on the feature similarity of  $p$  to other patches. This is based on an implicit assumption that features of patches within the same object should be more similar to each other than to features of other objects or of the background. However, this assumption is often not satisfied, since different parts of the same object may have very different appearances, and some object patches may be more similar to the background or other object patches. This fact limits learning efficiency and also the generalization ability on both in distribution and out of distribution samples. In addition, ViTs are susceptible to

learning “shortcuts” [27] where rather than capturing the object focused semantic meaning of an image, they capture spurious correlations with the background or other image artifacts. For example, if all training images show a fox in the forest, then a fox on a street may not be recognized. Currently, this problem is alleviated with a large number of training images via datasets such as ImageNet and heavy data augmentation. The hope is that the fox will appear on a large variety of backgrounds, but the assurance of this fact comes only from a large number of images, and it is hard to guess what other anomalies may be hidden in the training images.

Our key contribution is to limit the attention of patches to patches of the same object class only in a learned way. It can be viewed as refocusing the attention on relevant image parts. As demonstrated in [31], such an approach can lead to significant performance improvement. However, the focal modulation in [31] is done outside the transformer framework, and it does not include any inductive bias to focus on patches of the same object class, as proposed here.



**Fig. 1.** We restrict learning attention to objects of the same class.

We illustrate our idea on an example image in Fig. 1. We propose to limit the attention of the green patch  $p$  inside the dog to patches inside the green mask. Hence, the red patch in the background and the blue patch inside the cat in the blue mask are excluded from computing the new weighted representation of the green patch. Furthermore, our proposed restriction on the patch attention is not hard coded but learned. This is achieved by adding a new branch to selected attention layers that computes an auxiliary loss called object focused attention (OFA) loss. To train ViTs with the proposed semantically focused attention, we use datasets with semantic segmentation masks. Luckily there exists a plethora of such datasets like the MS COCO dataset or PASCAL VOC 2012 dataset.

In the absence of segmentation masks out of the box, we note the ability to generate pseudo-segmentation masks via general purpose segmentation models such as the Segment Anything Model (SAM) [13].

The proposed restriction of attention to patches within the same object allows transformers to gain a better understanding of configurational (or holistic) object shapes since attention is trained to be learned within patches of the same object class, hence the background is largely ignored. This also means better generalization to out-of-distribution (OOD) images. We present an experimental evaluation to demonstrate these facts on multilabel classification tasks. As our baseline model, we use the Musiq Transformer [12] and also show results with strong out-of-distribution performance with the standard ViT [5]. We chose Musiq Transformer due to its 2D positional encoding that is suitable for multiscale image representation. The original Musiq Transformer is developed for image quality assessment, but we adopt it for other downstream tasks such as multilabel classification. We note that our proposed OFA branch can be easily added to the self-attention layer of any vision transformer variant.

## 2 Object Focused Attention

As outlined in Sect. 1, our key idea is to limit the attention of patches to patches of the same class. Here we introduce our formal framework to implement this idea.

For ViT and its variants, an input image  $I$  is first divided into  $N$  disjoint square patches  $\mathcal{P} = \{p_1, \dots, p_N\}$  of a fixed size. For simplicity of presentation, we focus on a single encoder layer of Musiq [12] with one head. Let  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  be the set of input tokens representing the patches that were obtained by the previous layer, where each token is a row feature vector  $\mathbf{x}_i \in \mathbb{R}^d$ . Let  $\mathbf{X} \in \mathbb{R}^{N \times d}$  be the matrix obtained by stacking vectors  $\mathbf{x}_1 \dots \mathbf{x}_N$ . The scaled attention module of this layer first linearly projects the patch tokens to query, key, and value matrices  $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{d \times N}$ , given by  $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$ ,  $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ ,  $\mathbf{V} = \mathbf{X}\mathbf{W}_V$ , where  $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$  are learnable parameter matrices.

Next, we compute the attention weight matrix  $\mathbf{A}$  that reflects the similarity between the patches:

$$\mathbf{S} = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} \quad \text{and} \quad \mathbf{A} = \text{softmax}(\mathbf{S}) \in \mathbb{R}^{N \times N}. \quad (1)$$

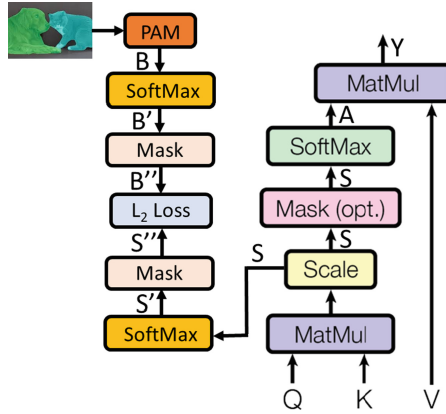
We call matrix  $\mathbf{S}$  a scaled pre-attention matrix. The  $i$ -th row of  $\mathbf{S}$  is denoted as  $\mathbf{s}_i \in \mathbb{R}^{1 \times N}$ , and it indicates the attention of patch  $i$  to all other patches.

Finally, the output matrix is obtained as  $\mathbf{Y} = \mathbf{A}\mathbf{V} \in \mathbb{R}^{N \times d}$ , where each row  $\mathbf{y}_i$  of matrix  $\mathbf{Y}$  is a new representation of patch  $\mathbf{x}_i$  as the sum of vectors in  $\mathbf{V}$  weighted by  $i$ -th row  $\mathbf{a}_i$  of attention matrix  $\mathbf{A}$ . The new representation of the  $i$ -th patch token is a weighted sum of all patch tokens.

The right branch of the diagram in Fig. 2 illustrates this process, which is the standard attention computation as proposed in [28]. The left branch of the diagram in Fig. 2 illustrates the proposed object focused attention (OFA) that



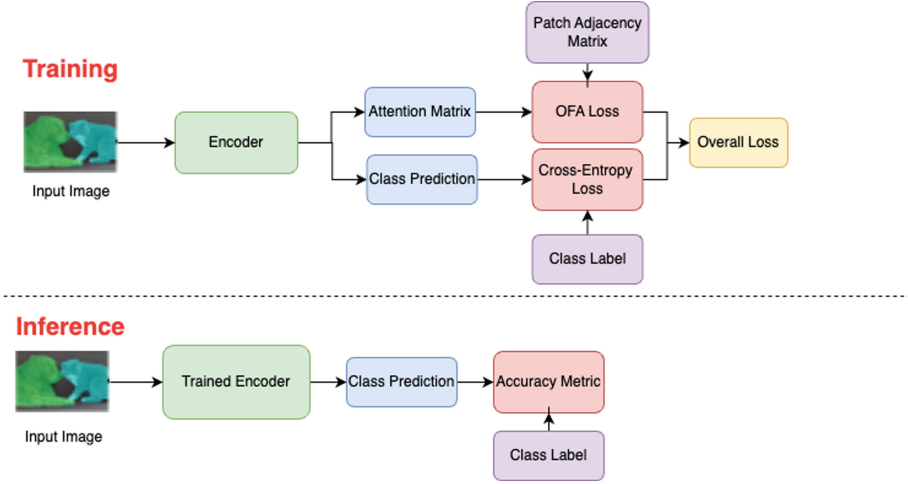
aims at training matrix  $\mathbf{S}$  to resemble a binary matrix  $\mathbf{B}$  representing a focus on patches within a given object. The left branch is devoted to computing OFA loss. The matrix  $\mathbf{B}$  and the process of computing OFA loss are defined below (Fig. 3).



**Fig. 2.** The proposed object focused attention (OFA) as an extension of self-attention. The arrows are labeled with the input/output matrices. The right part of the diagram is based on the original self-attention paper [28]. The left branch computes the OFA loss. The patch adjacency matrix (PAM) module is used to compute the patch adjacency matrix  $\mathbf{B}$ , which is then compared to the pre-attention matrix  $\mathbf{S}$ .

Let  $\mathcal{R} = \{R_1, \dots, R_r\}$  be a semantic segmentation of image  $I$  into a set of disjoint regions (object masks) such that their union covers the whole image. We also assume that patch  $p_i$  is contained in or intersects region  $R_j$ . Our training procedure seeks to reduce the attention values of patches disjoint with region  $R_j$  to zero in the pre-attention vector  $\mathbf{s}_i$ . We note that simply setting these values to zero for the training image  $I$  will not generalize to test images for which no segmentation masks are given. Therefore, we propose to learn this behavior by incorporating an auxiliary loss function to focus the attention of patch  $i$  only on patches that also intersect region  $R_j$ . For this, we define a patch attention matrix (PAM)  $\mathbf{B}$ , which is a binary  $N \times N$  matrix. Ones in row  $\mathbf{b}_i$  of  $\mathbf{B}$  represent patches that intersect the same object mask as patch  $i$ . Formally,  $\mathbf{b}_{ik} = 1$  if both patches  $p_i$  and  $p_k$  intersect the same object mask and zero otherwise. We use here a simplified notation for clarity of presentation. In particular, patch  $p_i$  may intersect more than one object mask  $R_j$ , in which case more regions need to be considered. To handle overlap patches, we use a simple heuristic where if any part of a patch is part of an object, it is considered an object patch.

Then we apply row-wise softmax to  $\mathbf{B}$  and obtain  $\mathbf{B}' = \text{softmax}(\mathbf{B})$ . Since we want the patch cross attention to focus on foreground objects, we mask all rows in  $\mathbf{B}'$  that represent background patches. We denote the new matrix  $\mathbf{B}''$ .



**Fig. 3.** Data flow showing differences in training and inference. OFA is shown explicitly as a training time method and thus can be used without any segmentation labels during inference.

Similarly, we compute row-wise softmax to obtain  $\mathbf{S}' = \text{softmax}(\mathbf{S})$ . Followed by setting to zero (masking) all rows in  $\mathbf{S}'$  that represent the background patches. The resulting matrix is denoted with  $\mathbf{S}''$ . We use matrices  $\mathbf{S}''$  and  $\mathbf{B}''$  to define the object focused attention (OFA) loss as their  $L_2$  distance:

$$\mathcal{L}_{OFA} = \|\mathbf{S}'' - \mathbf{B}''\|_2. \quad (2)$$

This process is graphically illustrated in the left part of the diagram in Fig. 2. We call the transformer trained with this auxiliary OFA loss **OFAMusiq**.

In order to explain the intuition behind OFA loss, let us assume that row  $i$  of  $\mathbf{B}$  represents an object patch and has  $k$  ones, meaning there are  $k$  other patches that intersect the same region. Then  $\text{softmax}(\mathbf{B})$  maps the ones in row  $i$  of  $\mathbf{B}$  to  $1/k$  in  $\mathbf{B}'$ , and the same values will remain in  $\mathbf{B}''$ . Hence the  $L_2$  distance between rows  $i$  of  $\mathbf{B}''$  and  $\mathbf{S}''$  pushes patch  $i$  to pay equal attention to the other  $k$  patches of the same object and zero attention to all other patches. With reference to Fig. 1, OFA loss forces the green patch inside the dog to pay attention only to patches inside the green dog region.

Moreover, since the sum of each row of  $\mathbf{B}''$  is one, the contribution of each patch to OFA loss is equal. This means that a patch  $i$  that belongs to a small object, and hence has fewer neighbors in its attention graph (fewer ones in  $i$  row of  $\mathbf{B}$ ) is equally important as patches that belong to large objects.

The proposed OFA loss can be placed at any layer or at several layers at the same time. In Sect. 6, we explore options for the best placement of the OFA loss. Our overall loss function can be summarized as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{OFA}, \quad (3)$$

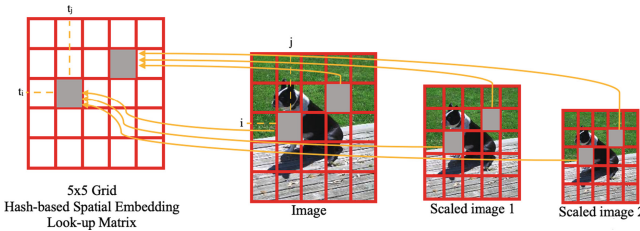
where  $\mathcal{L}_{task}$  is a task-dependent loss, e.g., cross-entropy for classification, and  $\alpha$  is a hyperparameter that balances the two loss functions.

### 3 Self-supervised Option with MAE

Our method uses datasets with semantic segmentation masks to train vision transformers with the proposed semantically focused attention. While there exist many such datasets such as the MS COCO dataset, PASCAL VOC 2012, or PACO [20], they are relatively small, so we explore the setting with self-supervision which is useful in learning representations for low-data domains. For this, we show experiments where we integrate OFA and Musiq Transformer with Masked AutoEncoder (MAE) [11].

MAE uses self-supervised learning masking, where certain patches of an image are masked, and the model is tasked with predicting the original content within those masked regions. This approach encourages the model to learn meaningful representations by leveraging contextual information from the surrounding visual context. The advantages of MAE lie in its ability to capture rich contextual dependencies and learn robust visual representations. Training the model to predict masked regions forces the model to understand and utilize the relationships and patterns present in the small amount of labeled data.

To our knowledge, we are the first ones to extend MAE to multiscale masking by utilizing Musiq positional encoding. Instead of performing masking directly on image patches, we propose to perform masking on the cells of the reference grid, which is then carried to tokens of images of different scales using a simple geometric mapping of the cell grids to image patches, see Fig. 4. This mapping is used by Musiq for positional encoding, but we extend it to also guide the masking process.



**Fig. 4.** The multiscale masking is computed by masking the grid cells (left) and carrying over the masked cells to image patches that correspond to those cells.

### 4 Adjacency Regularization

Another way to view our OFA loss is through the lens of “adjacency regularization” by enforcing a penalty on allowed states of connectivity. Vanilla transformers such as ViT are known for their  $O(N^2)$  quadratic complexity with respect to

attention computation over the number of input patches. Cast in the language of graphs, this is a complete graph (with self-loops) where every pair of vertices is connected via an edge producing  $\frac{N \cdot (N+1)}{2}$  edges of order  $O(N^2)$ . The adjacency matrix,  $\mathbf{B}$ , for such a graph can be described as  $\mathbf{B}_{ij} = 1 \forall i, j \in \{0, \dots, N-1\}$ , where  $N$  is the number of vertices in the graph, or in the case of ViT, input patches. By restricting the attention of each object patch to only patches of the same object, we significantly reduce the number of edges in the attention graph represented by matrix  $\mathbf{B}$ . In particular for MS COCO [15], purely object based connectivity creates a roughly 80% reduction in the number of edges. Only 20.7% of edges from the standard fully-connected attention are used. The underlying motivation behind training to decrease connectivity is to encourage a more parsimonious attention matrix which is robust to spurious correlations and instead can focus on semantic object information [14]. We show empirically in Sect. 6 that our model achieves such robustness.

## 5 Related Work

### 5.1 Transformers and Self-Attention

The transformer’s [28] self-attention mechanism offers a way for allowing every token to model information over every other token. ViT [5] adapted the transformer for computer vision by converting an image to a set of patch tokens and then using the standard transformer blocks.

There has been a considerable amount of work related to improving the self-attention mechanism and augmenting the inductive biases in vision transformers. One line of research has focused on modifying the self-attention mechanism to better capture spatial information in images. For example, [4] suggests using a mixture of local and global tokens in the input embedding to improve the model’s ability to capture both local and global information in the image. Swin transformer [18] utilizes a hierarchical structure analog to Convolutional Neural Networks (CNNs) to improve ViT performance. Learning of attention has been considered in [19], where it is applied to rectangular windows of patches. Since the size of the windows is learned, the approach is called window-free multi-head attention. In contrast to our work, all these approaches do not explicitly utilize object mask knowledge in restricting or restructuring self-attention. Moreover, many of them add computational overhead at the time of inference while our approach keeps the original structure of the self-attention layer during the inference.

### 5.2 Holistic Shape Representation

According to [1], objects have both local and configural shape properties. Local shape properties can be important for recognition. For example, ears alone may be sufficient to identify a rabbit but often are not discriminative enough. A configural shape property is a function not just of one or more local features (parts) but also of their arrangement meaning it provides a holistic shape representation.

Vision transformers, like other deep learning models, can learn to attend to different features of an image, including both texture and shape. However, it has been observed that their attention is more focused on texture than shape, in particular, they fail to capture the configurational nature of shapes in images, which means they are not able to adequately learn a holistic shape representation [1]. There have also been several studies on CNNs that demonstrated that they tend to attend more to texture than shape in natural images, e.g., [2, 9, 10]. We demonstrate in Sect. 6 that the proposed refocusing of attention within objects contributes to a better understanding of the holistic shape of objects.

### 5.3 Multi-label Classification

In many classification tasks, class labels are mutually exclusive such as when an image contains just one object. In multi-label classification, we predict mutually non-exclusive class labels, such as when an image may contain more than one object or concept. Multilabel classification is a challenging problem in computer vision due to the high dimensionality of the label space and potential correlations between labels. The label space can contain a large number of labels, and each label can be associated with multiple instances in the dataset. Furthermore, the labels can be highly correlated, meaning that the presence of one label in an image can increase the likelihood of other labels being present as well.

One of the first transformer networks applied to multilabel classification is [3], where windows partitioning, in-window pixel attention, and cross-window attention are used for improving the performance of multi-label image classification tasks. One of the best-performing multilabel classification method is ADDS [30], where ADDS stands for Aligned Dual moDality ClaSsifier. It includes a dual-modal decoder that performs alignment between visual and textual features. In contrast, we only use visual features.

## 6 Experimental Evaluation

Across our experiments, we use both single-scale and multi-scale MUSIQ transformers [12], denoted MUSIQ-single and MUSIQ-multi. The single-scale resizes images so that the longer side has length 512 while preserving the aspect ratio (ARP). The multi-scale uses the full-size image and two ARP resized inputs 384 and 224. It, therefore, uses three-scale input. In addition, we investigate the influence of self-supervised learning using MAE masking as a further enhancement of our methods. We also show that OFA is much more robust to background perturbations than standard ViTs by evaluating on our Stable Diffusion inpainted dataset. We use  $\alpha = 0.7$  across our experiments unless otherwise stated. Finally, we present an interesting finding via patch shuffling showing that ViTs don't grasp the overall shape of objects well compared to models equipped with OFA.

## 6.1 Multi-label Classification on MS-COCO and Pascal Voc2012

MS COCO (Microsoft Common Objects in Context) is a large-scale image recognition dataset containing 80 different object categories. Multilabel classification uses the same train/val splits as for the object detection task. The training set contains 118,287 images with annotations, while the validation set contains 5,000 images, which are used for testing. All the training images also contain semantic segmentation masks so that we can use them in our framework. We use the standard definition given by the COCO dataset of *thing* and *stuff*. From the COCO homepage we quote: “Things are objects with a specific size and shape, that are often composed of parts. Stuff classes are background materials that are defined by homogeneous or repetitive patterns of fine-scale properties, but have no specific or distinctive spatial extent or shape.” Put simply the COCO dataset defines segmentation masks directly for object classes and background classes.

Pascal VOC 2012 [6] contains objects grouped into 20 classes. The standard train/val set for the multilabel image classification/detection task has 11,540 images. However, since we need semantic segmentation masks, we train on train/val 2,913 images that are usually used for the image segmentation task. We test on the standard Pascal VOC 2012 test set composed of 10,991 images. Following other methods, we use mean average precision (mAP) in evaluating multilabel classification performance.

We experiment with computing the OFA loss over multiple attention layers of MUSIQ, which has 14 attention layers. Table 1 compares two settings for positioning the OFA loss: at the first and last layers [1, 14] and at layers [1, 7, 14]. Since placing OFA loss at layers [1, 7, 14] performs the best across all the settings, this model is used in all our further experiments. As for our weighting schema, we progressively weight the contributions of each attention block with later layers getting more weight with a factor of 0.9. The loss at layers [1, 7, 14] is weighted as:

$$OFA_{total} = \frac{1}{3}(0.9 \cdot OFA_{14} + 0.9^2 \cdot OFA_7 + 0.9^3 \cdot OFA_1) \quad (4)$$

The loss at layers [1, 14] is weighted as:

$$OFA_{total} = \frac{1}{2}(0.9 \cdot OFA_{14} + 0.9^2 \cdot OFA_1) \quad (5)$$

Table 2 shows multilabel classification results of MUSIQ transformer trained on MS COCO. We evaluate it on MS COCO and on Pascal VOC2012. The results on Pascal VOC2012 can be interpreted as zero-shot since do not train the model on this dataset and instead just fine-tune a classification head. We only benefit from the fact that the 20 classes of Pascal VOC2012 are a subset of the 80 classes of MS COCO. However, these datasets are composed of disjoint images, and MS COCO images are very different from Pascal VOC2012 images. Hence the excellent performance of MUSIQ-multi + MAE + OFA gives an initial result showing out-of-distribution (OOD) generalization ability of our approach.

**Table 1.** mAP multi-label classification results for placement of the OFA across layers. Placing OFA loss at layers [1, 7, 14] performs the best across all MUSIQ settings and so is used in further experiments. We add ViT and note that we use layer 12 instead of 14 as ViT-Base has 12 layers.

Methods	MS COCO		PASCAL VOC2012	
	[1,14]	[1,7,14]	[1,14]	[1,7,14]
MUSIQ-single + OFA	88.3	89.0	87.8	88.4
MUSIQ-multi + OFA	89.4	89.9	89.3	90.1
MUSIQ-single + MAE + OFA	91.3	91.7	90.8	91.5
MUSIQ-multi + MAE + OFA	91.6	92.1	91.2	91.9
ViT-Base + OFA	88.2	89.0	-	87.8

**Table 2.** mAP multilabel classification results on the MS COCO and Pascal VOC2012 datasets. All models are trained and evaluated on MS COCO. They are then applied on Pascal VOC2012 without any finetuning besides the linear head.

Methods	MS COCO	zero-shot VOC2012
ViT-Base	86.6	81.7
ViT-Base + OFA	87.3	87.8
MUSIQ-single	87.5	89.7
MUSIQ-multi	88.0	90.2
MUSIQ-single + OFA	89.0	90.9
MUSIQ-single + MAE	89.7	92.3
MUSIQ-multi + OFA	89.9	93.2
MUSIQ-multi + MAE	91.6	93.6
MUSIQ-single + MAE + OFA	91.7	94.7
MUSIQ-multi + MAE + OFA	<b>92.1</b>	<b>95.4</b>

In Table 3, we compare our methods to other multi-label classification methods on MS COCO, most with more complex architectures. We find that our method which adds an auxiliary loss to MUSIQ transformers outperforms other SOTA methods. We do not compare against multimodal methods such as [23, 30] since we only use visual features.

In Fig. 5, we visualize the final-layer attention maps of the baseline MUSIQ and MUSIQ + OFA for some test examples. We find that MUSIQ + OFA qualitatively attends to object shapes more consistently and produces reasonable segmentation maps in comparison to MUSIQ. This finding is consistent across small-single label images, large single-label images, multi-label images, and multi-label multi-object images. MUSIQ often attends more greatly to the background and finds spurious correlations through attention while the OFA loss has a significant

**Table 3.** Comparison to other methods on MS COCO. Our approach is SOTA against other methods and a MUSIQ-multi baseline. Combining multi-scale training and OFA gives better performance even at lower resolutions.

Methods	Resolution	mAP
IDA-R101 [16]	576	86.3
TResNet-XL [22]	640	88.4
TResNet-L-V2 [21]	640	89.8
MITr-XL [3]	384	90.0
IDA-SwinL [16]	384	90.3
Q2L-SwinL [17]	384	90.5
MLD-TResNet-L-AAM [26]	640	91.3
Q2L-CvT [17]	384	91.3
MUSIQ-multi	(full,384,224)	88.0
MUSIQ-multi + MAE + OFA	(full,384,224)	<b>92.1</b>

**Table 4.** Results of multilabel classification over 20 classes on Pascal VOC2012.

Method	mAP
VGG-16 [25]	79.3
Swin-B [18]	84.9
DeiT-B [29]	83.0
ViT-B [5]	81.7
PF-DL DL [7]	92.4
MCAR [8]	94.3
MUSIQ-multi	90.2
MUSIQ-multi + MAE + OFA	<b>95.4</b>

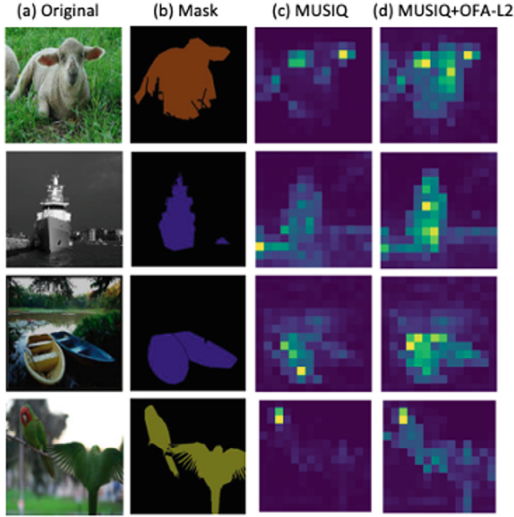
impact in focusing the attention computation on objects and greatly reducing attention to the background.

Table 4 compares the performance of zero-shot MUSIQ-multi + MAE + OFA (trained on MS COCO), to recent SOTA transformers: ViT-B [5], Swin (Swin-B) [18], DeiT with iRPE-K (DeiT) [29], PF-DL DL [7], MCAR [8] and to VGG-16 [25]. Our model exhibits the best performance and significantly outperforms the other methods (Table 6).

## 6.2 Out-of-Distribution Background Corruption with Stable Diffusion

In Fig. 6, we show selected examples of our new dataset for evaluation of OFA on OOD samples with adversarially corrupted backgrounds. We use Stable Diffusion inpainting [24] to replace backgrounds in each of the MS COCO test images with





**Fig. 5.** Comparison of attention maps of proposed MUSIQ + OFA and baseline MUSIQ.

**Table 5.** mAP results on MS COCO test data with background in-painted by Stable Diffusion [24]. We show the performance on the original test set and the degradation on our inpainted dataset. The OFA model is more robust to background perturbations. The result implies that OFA is more focused on learning semantic information about the objects rather than spurious correlations to the background.

Base Model	Resolution	Baseline ViT	ViT+OFA
ViT-Base-Patch16 (1k)	224	73.9 (-7.0)	<b>78.6 (-2.2)</b>
ViT-Base-Patch16 (21k)	224	73.6 (-9.3)	<b>81.7 (-2.2)</b>
ViT-Large-Patch16 (21k)	384	79.0 (-6.9)	<b>83.7 (-3.0)</b>

five new background categories: ocean, desert, forest, meadow, and beach. We use the mask information for each image to set boundaries for parts of the image that are inpainted. We inpaint the background of each image while leaving the object area unaltered, effectively superimposing each object onto a new background. To decide on the inpainting domain we use the simple prompts to guide the diffusion process. We use 5 prompts for each validation image resulting in an overall set of  $5 \times 5000 = 25,000$  images. We then test models trained on MS COCO without any finetuning. Table 5 clearly shows the robustness of OFA to OOD images with respect to background perturbations. We find that ViTs are susceptible to background perturbations showing a significant decrease in performance while the OFA model is more robust to background swapping.

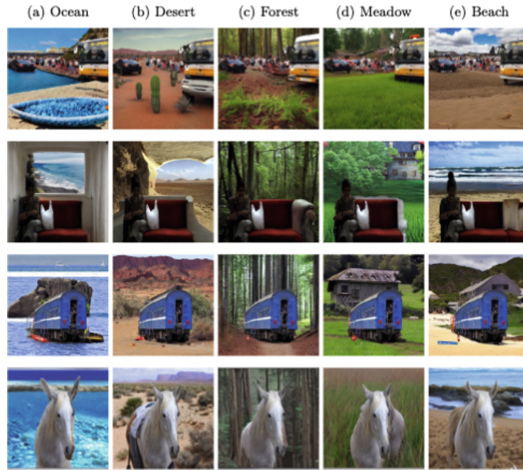


Fig. 6. Example images generated by Stable Diffusion inpainting on MS COCO.

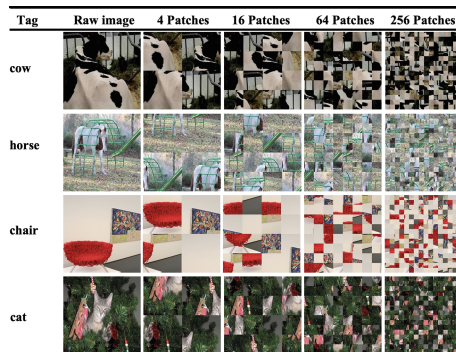


Fig. 7. Example shuffle operation applied to a varying number of patches. For humans the objects in a shuffled grid with 4 patches already seem unrecognizable.

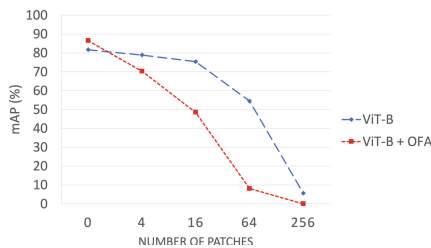


Fig. 8. The mAP over 20 classes on PASCAL VOC2012 when patches are shuffled. While the classification performance of ViT + OFA drops significantly, those of ViT hardly drops.

**Table 6.** Ablation of computing OFA loss on multiple attention blocks in ViT+OFA using the ViT-Base-Patch16 (21k) on a subset of MS COCO.

OFA at Different Layers (40% data)	1	2	3	4	5	6	7	8	9	10	11	12	mAP
[12]												✓	83.5
[1]	✓												83
[1,12]	✓											✓	83.6
[1,6,12]	✓					✓						✓	83.7
[1,3,7,10,12]	✓			✓			✓			✓		✓	<b>84.0</b>
[1,3,5,7,9,11]	✓	✓		✓		✓	✓		✓		✓	✓	83.7
[all]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	83.6

### 6.3 Learning Shape Representations over Textures

We demonstrate that the arrangement of object parts is not well represented by a standard ViT and is aided by using OFA. We divide an input image into patches by imposing a grid structure of different sizes and then randomly permute the position of patches. Figure 7 shows samples of this shuffle operation applied to PASCAL VOC 2012 images [6]. As illustrated by the blue dashed curve in Fig. 8, the multilabel classification performance of ViT remains nearly constant if 4 patches and 16 patches are permuted. However, as can be seen in Fig. 7, already the objects in the images with 4 permuted patches seem unrecognizable to a human. If ViT possessed an understanding of the configural shape, we should see a significant performance drop. In contrast, the performance of ViT + OFA drops significantly (red dashed curve). This demonstrates that it gained at least a rough understanding of configural object shapes due to the object-focused attention loss. We used ViT as the baseline model in this experiment to eliminate any influence of multi-scale and aspect ratio preserving since ViT takes a single-scale, square image of size  $256 \times 256$  as input.

## 7 Discussion and Future Work

We introduce a simple yet effective method for object-centered learning in the vision transformer framework. The proposed object focus attention loss is easily integrated into the self-attention module. Our trained model does not introduce any computational overhead at inference and still outperforms SOTA transformers. Moreover, it generalizes better to out-of-distribution examples and corrupted examples with respect to background and object shape. Finally, we show SOTA results when our approach is combined with multi-scale representation and MAE, offering a potential avenue for more exploration. We are interested in scaling our method to larger data using models that generate pseudo-segmentation masks such as SAM. We will explore this option in our future work. As shown in [2, 9, 10], deep learning models tend to focus on texture rather than on the shape of objects. Our experimental results demonstrate that the proposed refocusing

of attention on segmentation masks contributes to a better understanding of holistic object shapes. We speculate that this fact makes our model more robust to adversarial attacks. In order to refine the learned attention, we will also consider learning attention based on instance segmentation as well as on panoptic segmentation data.




## References

1. Baker, N., Elder, J.: Deep learning models fail to capture the configural nature of human shape perception. *iScience* **25** (2022)
2. Baker, N., Lu, H., Erlikhman, G., Kellman, P.J.: Deep convolutional networks do not classify based on global object shape. *Comput. Biol.* **14**(12) (2018)
3. Cheng, X., et al.: Mltr: multi-label classification with transformer. *IEEE Int. Conf. on Multimedia and Expo (ICME)* (2022)
4. Chu, X., et al.: Twins: revisiting spatial attention design in vision transformers. *NeurIPS* (2021)
5. Dosovitskiy, A., et al.: An image is worth  $16 \times 16$  words: transformers for image recognition at scale. *ICLR* (2021)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>
7. Gao, B.B., Xing, C., Xie, C.W., Wu, J., Geng, X.: Deep label distribution learning with label ambiguity. *IEEE Trans. Image Process.* **26**(6), 2825–2838 (2017)
8. Gao, B.B., Zhou, H.Y.: Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Trans. Image Process.* **30**, 5920–5932 (2021)
9. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR* (2019)
10. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. *NeurIPS* (2018)
11. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners (2022)
12. Ke, J., Wang, Q., Wang, Y., Milanfar, P., Yang, F.: Musiq: multi-scale image quality transformer, pp. 5148–5157 (2021)
13. Kirillov, A., et al.: Segment anything (2023)
14. Liao, R., Schwing, A., Zemel, R., Urtasun, R.: Learning deep parsimonious representations. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc. (2016). <https://proceedings.neurips.cc/paper/2016/file/a376033f78e144f494bfc743c0be3330-Paper.pdf>
15. Lin, T.Y., et al.: Microsoft coco: Common objects in context (2014)
16. Liu, R., Huang, J., Li, T.H., Li, G.: Causality compensated attention for contextual biased visual recognition. In: *The Eleventh International Conference on Learning Representations*
17. Liu, S., Zhang, L., Yang, X., Su, H., Zhu, J.: Query2label: A simple transformer way to multi-label classification (2021). <https://doi.org/10.48550/ARXIV.2107.10834>, <https://arxiv.org/abs/2107.10834>
18. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows, pp. 10012–10022 (2021)

19. Ma, J., Bai, Y., Zhong, B., Zhang, W., Yao, T., Mei, T.: Visualizing and understanding patch interactions in vision transformer (2022). <https://arxiv.org/abs/2203.05922>
20. Ramanathan, V., et al.: Paco: Parts and attributes of common objects (2023)
21. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. arXiv preprint [arXiv:2104.10972](https://arxiv.org/abs/2104.10972) (2021)
22. Ridnik, T., Ben-Baruch, E., Zamir, N., Noy, A., Friedman, I., Protter, M., Zelnik-Manor, L.: Asymmetric loss for multi-label classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 82–91 (2021)
23. Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E., Noy, A.: Ml-decoder: scalable and versatile classification head (2021). <https://doi.org/10.48550/ARXIV.2111.12933>. <https://arxiv.org/abs/2111.12933>
24. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695 (June 2022)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR) (2015)
26. Sovrasov, V.: Combining metric learning and attention heads for accurate and efficient multilabel image classification. arXiv preprint [arXiv:2209.06585](https://arxiv.org/abs/2209.06585) (2022)
27. Tang, Y., et al.: Augmented shortcuts for vision transformers (2021)
28. Vaswani, A., et al.: Attention is all you need (2017)
29. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer, pp. 10033–10041 (2021)
30. Xu, S., Li, Y., Hsiao, J., Ho, C., Qi, Z.: A dual modality approach for (zero-shot) multi-label classification (2022)
31. Yang, J., Li, C., Dai, X., Yuan, L., Gao, J.: Focal modulation networks (2022)



# Stereographic Projection for Embedding Hierarchical Structures in Hyperbolic Space

Shangyu Chen<sup>1</sup>✉, Xiaohao Yang<sup>1</sup>, Pengfei Fang<sup>2</sup>,  
Mehrtash Tafazzoli Harandi<sup>1</sup>, Dinh Phung<sup>1</sup>, and Jianfei Cai<sup>1</sup>

<sup>1</sup> FIT-DSAI, Monash University, Exhibition Walk, Clayton 3168, VIC, Australia

{shangyu.chen, Xiaohao.Yang, mehtash.harandi,  
dinh.phung, jianfei.cai}@monash.edu

<sup>2</sup> Southeast University, Nanjing 211189, China

fangpengfei@seu.edu.cn

**Abstract.** Hyperbolic geometry has emerged as a promising tool in diverse domains in deep learning. In this study, we concentrate on a key component of hyperbolic neural networks—the mapping from Euclidean space to hyperbolic space. We explore the problems and drawbacks of existing practices in this mapping, such as exponential mapping and projection methods constrained within the Poincaré ball. We emphasize that these methods rely entirely on supervised relationship data to capture hierarchical structure in hyperbolic space. The exponential mapping, which does not involve learning any parameters, functions more like a predefined activation function. This type of mapping does not convey any hierarchical structure information, making the computational cost of this mapping unnecessary. We propose a novel approach called Stereographic Projection Transition Mapping (SPTM). Leveraging the intrinsic properties of hyperbolic space, SPTM explicitly represents hierarchical structures present in the Euclidean space. By analytical mapping relationships in the Euclidean space, SPTM offers a more efficient and interpretable way to represent hierarchical structures in the Poincaré ball without the need for excessive supervision.

**Keywords:** Hyperbolic Neural Networks · Geometry of Neural Networks

## 1 Introduction

Since the introduction of hyperbolic neural networks [10], data characterized by tree structures and hierarchies, when processed in hyperbolic space, have found widespread applications in addressing various machine learning tasks [5, 10, 18, 20, 24, 26, 28]. The key difference between various Hyperbolic Neural Networks (HNN) and traditional machine learning lies in the embedding of data. When metrics such as Mean Squared Error (MSE), cosine similarity, or variance

are utilized, they are based on the assumption that the data is embedded in Euclidean space. For hierarchical data structures like graphs and trees, when comparing the distance between two nodes, calculating the distance between them and their common parent can better reflect the similarity between these nodes, as opposed to directly calculating the Euclidean distance between the coordinates of the two nodes. Hyperbolic spaces like the Poincaré ball, with its geodesics convex to the center [13], are capable of better fitting the path from one child node, through the parent node, to another child node. Hence, the distance function of hyperbolic space, as the line integral along the geodesic, is employed to fit the distance functions of hierarchical data structures such as graphs and trees.

In order to embed data into hyperbolic space, more specifically, usually within the Poincaré ball, the method predominantly utilized is exponential mapping. The data prior to mapping is presumed to be located on the tangent plane of the Poincaré ball, typically the tangent plane at the origin. The tangent plane is a Euclidean plane. With the two inverse mappings, the exponential mapping from the tangent plane to the Poincaré ball and the logarithmic mapping from the Poincaré ball to the tangent plane, the transformation between Euclidean space and hyperbolic space is achieved.

However, exponential mapping, as a mapping from the tangent plane to the Poincaré ball, can only ensure that the mapped data coordinates lie within the predetermined Poincaré ball. It does not contribute to the construction of a hierarchical structure of data. This paper will demonstrate that exponential mapping cannot map data that already has a hierarchical structure in Euclidean space into a tree structure. Then exponential mapping should be viewed merely as ensuring constraints within the Poincaré ball. The hierarchical structure is obtained by fitting the distance function of the Poincaré ball. This leads to the question: Are there direct mapping methods that facilitate the hierarchical structure? This paper will mainly focus on this question. We will propose a new method of embedding from Euclidean space to the Poincaré ball, starting from the geometric perspective of hyperbolic space, which we name as the *stereographic projection transition mapping (SPTM)*.

In Euclidean space, the assumption based solely on Euclidean distance can indeed facilitate hierarchical clustering. [19, 22]. In hyperbolic space, the relationship between clustering in Euclidean space and clustering in hyperbolic space has not been sufficiently considered. However, in hyperbolic word embedding models such as HNN and HyperMiner [27], one must rely on the supervision of relational datasets, such as WordNet [9]. This paper will demonstrate that unsupervised hyperbolic embedding is feasible with the assistance of Euclidean clustering. As a case in point, we have conducted experiments on word embedding and topic embedding on the Poincaré ball using an unsupervised topic model. However, it is important to emphasize that learning hierarchical word embeddings on the Poincaré ball with the help of a topic model is just one example method. The focal point of this paper is the stereographic projection transition mapping

(SPTM) method, which provides an analytic approach to extract hierarchical features from the Euclidean space and map them onto the Poincaré ball.

The *main contribution* of this paper is to bridge the gap between the embeddings of hierarchical structures in Euclidean space and the Poincaré ball. This is manifested in the following ways:

1. We propose a lightweight method stereographic projection transition mapping (SPTM) for embedding hierarchical structures from Euclidean space onto the Poincaré ball.
2. We identify the limitations of relying solely on the exponential mapping for obtaining hierarchical embeddings. It is essential to emphasize that such an approach, along with the projection based on norm constraints, is only viable when supervised with a relationship dataset.
3. We demonstrate the unsupervised implementation of hierarchical embeddings, which is distinct from existing methods that use contrastive loss based on relational datasets.

The findings obtained from experiments are worth mentioning but not listed as our main contributions. We found that the exponential mapping should be explicitly defined in a specific layer of the neural network, rather than expecting the neural network to inherently learn it. And we introduce a method for transferring existing embedded topic models from Euclidean embedding space to hyperbolic embedding space. Furthermore, we extend the application of SPTM to image reconstruction using the hyperbolic VQ-VAE model, providing experimental evidence that SPTM is also effective in the domain of vision tasks.

## 2 Background

### 2.1 Hyperbolic Neural Networks

Hyperbolic space, including hyperbolic neural networks [10], has garnered interest recently for its unique properties characterized by negative curvature. The Poincaré ball model and the Lorentz model are two common representations of hyperbolic space. The Poincaré ball model maps points within a unit ball, while the Lorentz model represents hyperbolic space as a hyperboloid.

The Lorentz model, also known as the Minkowski model, is a representation of hyperbolic space that is widely used in physics and relativity theory. It is defined by embedding hyperbolic space in a pseudo-Euclidean space of one additional dimension. In the  $(n + 1)$ -dimensional Lorentz model  $\mathbb{L}^{n+1}$ , hyperbolic space is described by the equation:

$$-x_0^2 + x_1^2 + x_2^2 + \dots + x_n^2 = -1 \quad (1)$$

where  $x_0, x_1, x_2, \dots, x_n$  are the coordinates of a point in the hyperboloid. The hyperboloid, which is a surface in the Lorentz model, represents the points in hyperbolic space and is defined by the same equation.



Within Poincaré ball, the Möbius addition operation is used to combine two points  $x$  and  $y$  and produce a new point  $z$ . It can be represented as:

$$\mathbf{z} := \mathbf{x} \oplus_c \mathbf{y} := \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2) \mathbf{x} + (1 - c\|\mathbf{x}\|^2) \mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2} \quad (2)$$

In hyperbolic space, distances are measured using the Poincaré distance, which quantifies the geodesic distance between two points. The exponential mapping and logarithm mapping operations allow for the transformation of points between Poincaré ball and the tangent plane of Poincaré ball, facilitating computations within the hyperbolic space. To define the distance on the Poincaré ball, we can use the following formula:

$$d_c(\mathbf{x}, \mathbf{y}) = (2/\sqrt{c}) \tanh^{-1}(\sqrt{c}\|-\mathbf{x} \oplus_c \mathbf{y}\|) \quad (3)$$

Here are the formulas for the exponential mapping and logarithm mapping in the Poincaré ball:

$$\text{exp}_{\mathbf{x}^c}(v) = \mathbf{x} \oplus_c \left( \tanh \left( \sqrt{c} \frac{\lambda_x^c \|v\|}{2} \right) \frac{v}{\sqrt{c}\|v\|} \right) \quad (4)$$

$$\text{log}_{\mathbf{x}^c}(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_x^c} \tanh^{-1}(\sqrt{c}\|-\mathbf{x} \oplus_c \mathbf{y}\|) \frac{-\mathbf{x} \oplus_c \mathbf{y}}{\|-\mathbf{x} \oplus_c \mathbf{y}\|} \quad (5)$$

In these equations,  $\mathbf{y}$  represent points on the Poincaré ball,  $v$  is a tangent vector,  $\mathbf{x}$  is the tangent point between the tangent space and the Poincaré ball, and  $c$  denotes the curvature.  $\|v\|$  represents the Euclidean norm of  $v$ , and  $\lambda_x^c$  is the Lorentz factor defined as  $\lambda_x^c = 1/\sqrt{1 - c\|\mathbf{x}\|^2}$ . The  $\tanh$  function is the hyperbolic tangent, and  $\tanh^{-1}$  is the inverse hyperbolic tangent function.

To practically integrate the geometry constraint, a transformation layer is used to map data embeddings from Euclidean space to hyperbolic space. This is achieved through a project function denoted as  $\text{Proj}(\cdot)$ , where  $\mathbf{z} = \text{Proj}(\mathbf{x})$  for  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{z} \in \mathbb{D}_c^n$ . One common approach, as seen in [4, 14], is to instantiate  $\text{Proj}(\cdot)$  using  $\text{exp}_0(\cdot)$ , followed by the following constraint:

$$\mathbf{z} = \Gamma(\mathbf{p}) = \begin{cases} \mathbf{p} & \text{if } \|\mathbf{p}\| < \frac{1}{\sqrt{c}} \\ \frac{1-\xi}{\sqrt{c}} \frac{\mathbf{p}}{\|\mathbf{p}\|} & \text{else,} \end{cases} \quad (6)$$

In this equation,  $\mathbf{p} = \text{exp}_0(\mathbf{x})$ , and  $\xi$  is a small value used to ensure numerical stability. Fang P. [8] argues that the above projection does not fully utilize the hyperbolic space, as it flattens every vector at the identity, limiting the ability to approximate the structure in hyperbolic spaces. To address this, an alternative approach applies the constraint in Eq. (6) only to the output of the feature extractor. By doing so, the network can optimize the encoding of the input data directly into the hyperbolic space [8, 16]. Previous deep hyperbolic networks have adopted a hybrid architecture, where a neural network first extracts feature embeddings of the input data in Euclidean space, and then a transformation

layer is used to obtain the hyperbolic embeddings. However, recent work [11] shows that this hybrid architecture can lead to vanishing gradients during back-propagation, limiting the network’s applicability. To mitigate this issue, Guo et al. propose a simple solution by clipping the Euclidean embeddings using the following formula:

$$\mathbf{p} = \Phi(\mathbf{x}) = \min\left\{1, \frac{r}{|\mathbf{x}|}\right\} \cdot \mathbf{x}, \quad (7)$$

Here,  $r$  is a hyper-parameter. After clipping, the embeddings are further projected to the hyperbolic space using the exponential mapping as  $\mathbf{z} = \exp_{\mathbf{0}}(\mathbf{p}) = \exp_{\mathbf{0}}(\Phi(\mathbf{x}))$ . This process bounds all embeddings within an  $r$ -radius sphere, ensuring they are located in an open ball of radius  $2r$  [11].

However, these methods are all about enforcing hard constraints on the domain, rather than being analytically tailored optimal approaches exclusively designed for hyperbolic spaces. We need a mapping that directly relates to representing hierarchical structures in hyperbolic space, which is what SPTM aims to achieve.

## 2.2 Topic Model

In the field of topic modeling, there are several classical models such as Latent Dirichlet Allocation (LDA) [3] and Probabilistic Latent Semantic Analysis (PLSA) [12]. These models summarize and model textual data to reveal the underlying topic structure in the text. The general framework of a topic model typically consists of two key matrices: the topic-word distribution matrix ( $\beta$ ) and the document-topic distribution matrix ( $\theta$ ). The topic-word distribution matrix ( $\beta$ ) is a  $K \times V$  matrix, where  $K$  represents the number of topics and  $V$  represents the size of the vocabulary. Each element  $\beta_{kj}$  of  $\beta$  represents the probability distribution of word  $j$  in topic  $k$ . It describes the association between topics and words and can be understood as the word distribution for each topic. The document-topic distribution matrix ( $\theta$ ) is a  $D \times K$  matrix, where  $D$  represents the number of documents. Each element  $\theta_{dj}$  of  $\theta$  represents the probability distribution of topic  $j$  in document  $d$ . It describes the association between documents and topics and can be understood as the topic distribution for each document. With these two matrices, a topic model is able to decompose and model text data, revealing the underlying topic structure. Specifically, given a document, the document-topic distribution matrix ( $\theta$ ) can be used to infer the importance of each topic within the document, and thus infer the document’s topic distribution. Similarly, the topic-word distribution matrix ( $\beta$ ) can be used to infer the importance of each word within each topic, and thus infer the word distribution for each topic.

In recent years, the introduction of word embedding technology has brought new developments to topic modeling. In these topic models, defining the metric space of word and topic embeddings becomes a pivotal concern. One recent example, the Embedded Topic Model (ETM) [6], integrates an embedding layer to learn distributed representations for each word, capturing the semantic information from words. A distinct variant, built entirely upon the metric of embeddings,

is the NSTM [29], which leverages the OT distance between topic embeddings and word embeddings to define the explicit coexistence relationship in a metric space. As the semantics of words and topics have hierarchical relations in nature, word embedding in Euclidean space limits the capability of most existing embedded topic models in capturing hierarchical semantics. More recently, researchers started to explore the feasibility of using hyperbolic spaces in topic modelling and HyperMiner [27] provides support and validation for using hyperbolic spaces in topic modelling.

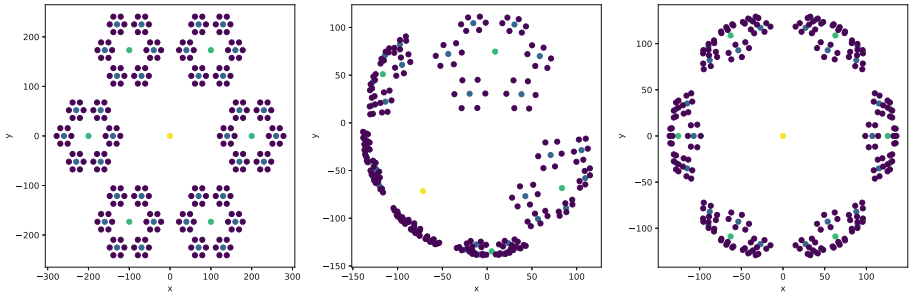
### 3 Stereographic Projection Transition Mapping

#### 3.1 Limitations of Exponential Mapping for Hierarchical Embeddings

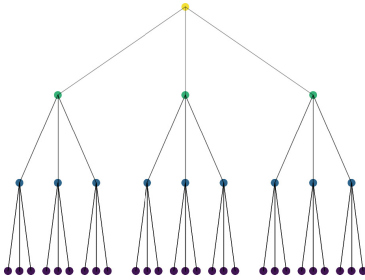
We commence with a toy experiment to identify the limitations of relying solely on the exponential mapping for obtaining hierarchical embeddings. We use a fractal tree to represent the hierarchical structure in 2D Euclidean space. The positions of each node in the hierarchical structure are calculated as coordinates, which include the x-coordinate, y-coordinate, and depth value, with the depth value indicating the number of steps from the root node. After combining the x and y coordinates into a new array, we set the curvature of the Poincaré ball (in this case, 0.00005) and apply the exponential map to these (x, y)-coordinates. The outcomes are then plotted in three sub-figures in Fig. 1: The first displays the original coordinates color-coded by depth value on the tangent space (Euclidean space), the second shows the coordinates after the exponential map  $\exp_{\mathbf{x}^c}(v)$  where the tangent point  $\mathbf{x}$  is not the origin, and the third shows the coordinates after the exponential map when the tangent point  $\mathbf{x}$  is the origin. All are color-coded by depth value.

In the context of representing hierarchical structures in hyperbolic neural networks, an ideal arrangement of embeddings in the Poincaré ball would have root nodes near the center and leaf nodes near the boundary. As shown in the Fig. 1, this arrangement cannot be accomplished solely by exponential mapping.

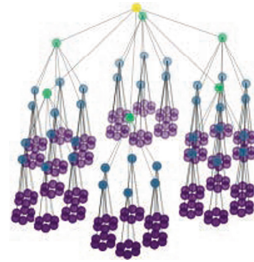
Alexandru demonstrated that when approaching the boundary, the embeddings approximate the upper half-plane hyperbolic model [13]. The center image in Fig. 1 provides an approximation of this situation. As seen from this figure, without the addition of dimensions and without a nonlinear mapping, the upper half-plane hyperbolic model has some effect in the lower left region of the middle image in Fig. 1, but it completely disregards the information from data points in the upper right region. In the upper-right corner, the hierarchical structure still closely resembles Euclidean space, where higher-level nodes are surrounded by subordinate nodes, rather than being closer to the center as expected in a hierarchical structure. For the points in the lower-left corner of the middle image in Fig. 1 and the points in the right image, we can observe a tendency for some subordinate nodes to approach the boundary compared to their corresponding higher-level root nodes. However, many subordinate points still appear closer to the center compared to the higher-level root nodes.



**Fig. 1.** Illustration of the fractal tree experiment for identifying the limitations of relying solely on the exponential mapping for obtaining hierarchical embeddings. Left: Euclidean coordinates on the tangent plane. Center: The origin of the fractal tree is shifted to the boundary and coordinates are then mapped exponentially on the Poincaré ball with a curvature of  $-0.00005$ . Right: The origin of the fractal tree remains at the center with coordinates mapped exponentially on the Poincaré ball with a curvature of  $-0.00005$ .



**Fig. 2.** hierarchical structure 1D to 2D



**Fig. 3.** hierarchical structure 2D to 3D

From Fig. 1 (left), it is observed that in this typical two-dimensional data, the hierarchical structure is expressed through different colors representing depth. Thus, an additional depth dimension is considered to embody the hierarchical structure of the data.

Figure 2 and Fig. 3 respectively represent the original 1-dimensional and 2-dimensional data, with an added depth dimension to reflect the hierarchical structure. Figure 3 visualizes the hierarchical structure from Fig. 1 in 3D.

### 3.2 Method: Stereographic Projection Transition Mapping

Firstly, it should be noted that our goal remains to map the data onto the Poincaré ball. Among the five common models of hyperbolic space [2] - the Lorentz (Hyperboloid) model, the Poincaré ball model, the Poincaré half space model, the Klein model, and the Hemisphere model - the Poincaré ball is the

ideal choice due to its isotropic properties, suitability for hierarchical structure’s geometric characteristics, and well-established computational libraries [1]. However, during the computation process, we utilize the Lorentz model, as it has a unique mapping with the Poincaré ball and lower computational complexity.

$$x'_i = \frac{x_i}{1 + x_0}, \quad \text{Hyperboloid to Poincaré Ball} \quad (8)$$

$$(x_0, x_i) = \frac{(1 + \sum_{i=0}^n x_i'^2, 2x'_i)}{1 - \sum_{i=0}^n x_i'^2}. \quad \text{Poincaré Ball to Hyperboloid} \quad (9)$$

In a  $R^{n+1}$  space, the coordinates of the hyperboloid are represented as  $(x_1, x_2, \dots, x_n, x_0)$ , while the coordinates of the Poincaré ball are represented as  $(x'_1, \dots, x'_n, 0)$ .

The overall strategy is to learn an additional dimension to reflect hierarchy, given the existing data embeddings in Euclidean space.

In the Poincaré ball  $\mathbb{B}^{n+1} \in \mathbb{R}^{n+1}$ , the extra dimension representing the hierarchy manifests as the distance to the origin, while data without a hierarchical structure are distributed on a hypersphere  $\mathbb{S}^n$  within the Poincaré ball. In the Lorentz model  $\mathbb{L}^{n+1} \in \mathbb{R}^{n+2}$ , the hierarchical dimension is represented by  $t$ , and data without a hierarchical structure lie on a hypersphere  $\mathbb{S}^n \in \mathbb{R}^{n+2}$  on the hyperboloid. To obtain the hypersphere  $\mathbb{S}^n$  on the hyperboloid, we stereographical project the original data from the Euclidean space  $\mathbb{R}^n$  onto the hypersphere  $\mathbb{S}^n$ .

Thus, the mapping from the Euclidean space  $\mathbb{R}^n$  to the Poincaré ball  $\mathbb{B}^{n+1}$  is composed of three parts.

- First, the stereographical projection from the Euclidean space  $\mathbb{R}^n$  to the hypersphere  $\mathbb{S}^n$ , as shown in Fig. 4,

$$\begin{aligned} &\text{For } (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, \\ &(x'_1, x'_2, \dots, x'_n, x'_0) = \\ &\left( \frac{2rx_1}{1 + \sum_{i=1}^n x_i^2}, \frac{2rx_2}{1 + \sum_{i=1}^n x_i^2}, \dots, \frac{2rx_n}{1 + \sum_{i=1}^n x_i^2}, \frac{-r + r \sum_{i=1}^n x_i^2}{1 + \sum_{i=1}^n x_i^2} \right), \end{aligned} \quad (10)$$

where  $r$  represents the radius of the sphere as a hyper-parameter, typically preset to 1.

- Second, the embedding of the hypersphere  $\mathbb{S}^n$  into the Lorentz model  $\mathbb{L}^{n+1}$ , the circle ( $\mathbb{S}^1$ ) in Fig. 4 is embedded onto the black circle on the hyperboloid in the Fig. 5,

$$\begin{aligned} &\text{For } (x'_0, x'_1, x'_2, \dots, x'_n) \in \mathbb{S}^n, \\ &(x''_0, x''_1, \dots, x''_n, h) = \left( x'_0 \frac{\sqrt{h^2 - 1}}{r}, x'_1 \frac{\sqrt{h^2 - 1}}{r}, \dots, x'_n \frac{\sqrt{h^2 - 1}}{r}, h \right) \end{aligned} \quad (11)$$

In this context,  $h$  can either be preset as a hyper-parameter or optimized during the training process. We will see an example in later sections where  $h$  is considered as a parameter to be optimized.

- Third, the mapping from the Lorentz model  $\mathbb{L}^{n+1}$  to the Poincaré ball  $\mathbb{B}^{n+1}$ . As shown in Fig. 5, the red samples on the black circle on the hyperboloid are mapped to the blue circle on the Poincaré disk in the x-y plane.

$$\begin{aligned} &\text{For } (x''_0, x''_1, \dots, x''_n, h) \in \mathbb{R}^n, \\ &(x'''_0, x'''_1, \dots, x'''_n) = (x''_0/(1+h), x''_1/(1+h), \dots, x''_n/(1+h)) \end{aligned} \tag{12}$$

The composition of these three mappings is referred to as Stereographic Projection Transition Mapping (SPTM).

To optimize  $h$  for each sample, the objective is to separate the samples on a specific  $S^n$  in the Poincaré ball. This separation is achieved by pulling the parent nodes towards the center of the ball, while pulling the subnodes towards the boundary of the Poincaré ball.

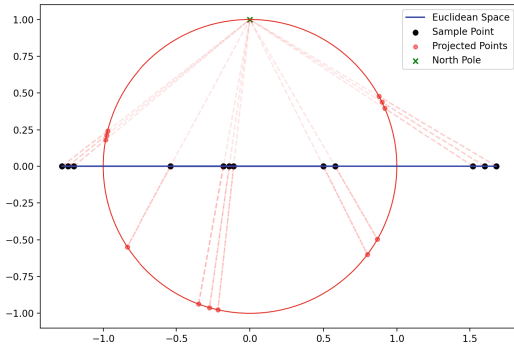


Fig. 4. Stereographic projection

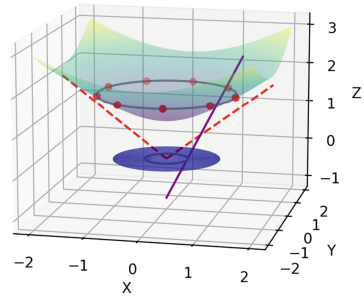


Fig. 5. Hyperboloid vs. Poincaré Disk Mapping

### 3.3 Optimization Algorithm for SPTM

The specific algorithm for obtaining hierarchical embeddings based on this mapping depends on the specific problem at hand. To optimize the parameter based on the distance in the Poincaré ball, we can use the distance from the samples to the center of the Poincaré ball, denoted as  $r' = 1/(1+h)$ , as the optimization parameter instead of  $h$ .

The general flow of the algorithm is as Algorithm 1.

We present a concrete method here based on fitting the distances between hierarchy levels on the Poincaré disk to the distances between hierarchy levels in Euclidean space. In Euclidean space, we perform clustering and calculate the distance matrix from nodes to cluster centers, denoted as  $M_e$ . After mapping the nodes and cluster centers using SPTM onto the Poincaré disk, the distance

---

**Algorithm 1.** Poincaré Ball Mapping and Optimization

---

- 1: **Input:** Tensor in Euclidean space,  $X$
  - 2: **Parameter:** Initial hyper-parameter,  $r'$ , Loss function based on Poincaré ball distance, Loss, Optimization algorithm, optimizer
  - 3: **Output:** Optimized  $h$ , Mapped tensor in Poincaré ball,  $X'''$
  - 4: Set the initial value for the hyperparameter  $r$
  - 5: // **SPTM to Poincaré ball**
  - 6: Project  $X$  to the Poincaré ball:
  - 7:  $X' = \left( \frac{2rX}{1+\|X\|^2}, \frac{-r+r\|X\|^2}{1+\|X\|^2} \right)$ ,
  - 8:  $X'' = \left( \frac{X' \cdot \sqrt{h^2-1}}{r}, h \right)$ ,
  - 9:  $X''' = X'' \cdot r'$
  - 10: **while** not converged **do**
  - 11:   Calculate loss  $L = \text{Loss}(X''')$
  - 12:   Calculate gradient of the loss with respect to  $h$ ,  $\text{grad}_{r'} = \frac{\partial L}{\partial r'}$
  - 13:   Update  $h$  using the optimizer with the gradient  $\text{grad}_{r'}$
  - 14:   Re-compute  $X'$ ,  $X''$ ,  $X'''$  using updated  $r'$
  - 15: **end while**
  - 16: **return**  $r'$ ,  $X'''$
- 

matrix  $M_p$  is computed. Then the optimization of  $h$  is based on minimizing the Mean Squared Error (MSE) between  $M_e$  and  $M_p$ . To ensure convergence, we need the elements in  $M_e$  to be not greater than **1.14** when we set the curvature to -1. Proof is available in the Appendix B. In the experiment, this boundary is satisfied by increasing the number of clusters as more clusters imply shorter distances within each cluster.

Specifically, we illustrate the algorithm using a topic model as an example. In this case, we rely on NSTM [29] and GLOVE [23] word embeddings. The advantage of using GLOVE is that it explicitly incorporates Euclidean distances during the training of word embeddings. We choose NSTM because it utilizes the Optimal Transport (OT) distance and requires the definition of explicit cost functions for topic and word embeddings. This distinguishes it from models like ETM (Embeddings from Language Models), which rely on probabilistic distribution divergences. Therefore, instead of using cosine similarity as in the original NSTM paper to measure distances between topics and words, we directly employ the Euclidean distance to obtain the topic distribution.

After training NSTM, we save the topic embeddings and the distance matrix between topic embeddings and word embeddings from it. We then perform Stereographic Projection Transition Mapping (SPTM) to map the topic embeddings and word embeddings to the Poincaré ball. Since the topic embeddings are derived from GLOVE word embeddings and their Euclidean distances, we can confidently consider the original topic embeddings and word embeddings to exist in Euclidean space. Next, on the Poincaré ball, we initialize the radius parameter  $r'$  for each embedding. We set the  $r'$  of the topic embeddings to be closer to the center of the ball, while the  $r'$  of the word embeddings is set to be relatively

farther from the center compared to the topics. We then recalculate the distance matrix between topic embeddings and word embeddings on the Poincaré ball to make it closely resemble the distance matrix obtained from NSTM. The pseudo code for the algorithm can be found in the Appendix C.

## 4 Experiment

In this section, we present the experimental setup and evaluation metrics used to assess the performance of our proposed hierarchical topic model based on Stereographic Projection Transition Mapping (SPTM-TM) compared to several hyperbolic mapping methods. We conducted extensive experiments on three benchmark text datasets: 20 News Groups (20NG) [15], Web Snippets (WS) [7], and Tag My News (TMN) [21].

One of the primary focuses of our experiments is to demonstrate the hierarchical visualization capability of SPTM-TM. In the existing research on hyperbolic space, the influence of topic modeling on optimization has limited effectiveness, and its strength lies more in the ability to visualize hierarchical structures. This visualization capability offers a certain level of interpretability to deep learning models. Therefore, we would like to emphasize the visualization capability of SPTM.

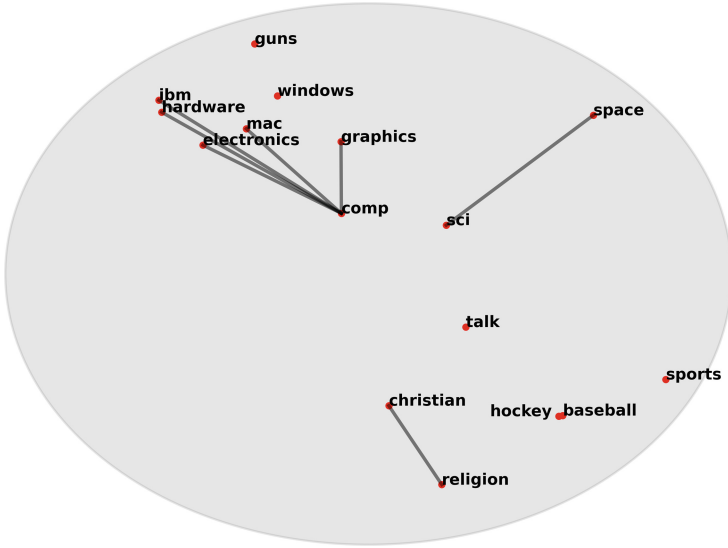
In this framework, the topic model has two sets of embeddings: the embeddings obtained from the original model (e.g. from NSTM. This part is based on NSTM) and the embeddings after mapping to the Poincaré ball. We can reconstruct the evaluation based on the topic word matrix  $\beta$  of the original model, which means that the mapped model will achieve the same results as the original model. Additionally, we can calculate the  $\beta$  based on the cosine similarity between the topic embeddings and word embeddings on the Poincaré ball for reconstruction purposes.

In addition, we conducted preliminary investigations into the applicability of SPTM to other modalities, such as image reconstruction. We conducted baseline experiments to evaluate the effectiveness of SPTM when applied to image reconstruction using a Vector Quantised-Variational AutoEncoder (VQVAE), with detailed results and methodology presented in Appendix F.

**Visualization.** We extract the hierarchical topics annotated in the 20News dataset [15] and retrieve their corresponding words from the database. The word embeddings of these hierarchical topics in the Poincaré ball are shown in Fig. 6.

Our approach, in contrast to supervised hyperbolic word embeddings like HNN [10] and HyperMiner [27], uses unsupervised methods NSTM and SPTM. Despite this, we still observe hierarchical topics in the embeddings, such as various computer-related topics in the 20News dataset, which are evident in the upper-left corner of the Fig. 6. However, these learned topics might not align perfectly with subjective annotations, as what is revealed here is the geometrically implicit hierarchical structure within the data. This structure is related to, but not exactly the same as, the hierarchical structure understood by humans.





**Fig. 6.** Visualization of Word Embeddings in Poincaré Ball

For example, while humans may subjectively perceive “religion” to be a more abstract concept and closer to the root node than “christian”, the data might reveal that “christian” occurs more frequently and have broader connections, thereby positioning it closer to the root node.

In Appendix D, We also visualize the two-dimensional word embeddings learned by SPTM (different from Fig. 6, for clarity, we only provide words for named topics). For each of the three datasets, we select three clusters as representative learned topics. The visualizations clearly demonstrate that the distribution of topic embeddings in SPTM effectively preserves the semantic structure based on prior knowledge.

**Quantitative Results.** Baseline Methods and Settings: Based on NSTM, we applied exponential mapping to its word embedding and topic embedding, and also relied solely on the poincaré ball constraint defined in Eq. 6, and compared with SPTM-TM. We then computed several evaluation metrics commonly used in topic models. The aim here is to investigate whether, by granting hierarchical visualization capabilities to topic and word embeddings, there would be any impact on other aspects of the original model’s performance. Because the purpose of designing SPTM is to obtain a mapping that preserves hierarchical structures in Euclidean space and explicitly represent them without relying on additional supervised training data of relationships. Here, the hierarchical structures we aim to preserve are those of the topics and words trained under NSTM in the original Euclidean space. This topic model is unsupervised and therefore it is not supervised by a relationship dataset. Then we just require metrics regarding

the quality of the topic model to confirm that the quality of the original topic model has not been reduced.

We examined the performance of these mappings on other state-of-the-art neural topic models, namely ProLDA [25], ETM [6], and NVDM [17], as detailed in the Appendix E.

In evaluating the quality of topics, we focus on four metrics: Topic Diversity (TD), document classification accuracy, top-Purity, and Normalized Mutual Information (NMI), as detailed in Table 1. TD measures the uniqueness of words within topics, while accuracy, top-Purity, and top-NMI assess the effectiveness of document representations in classification and clustering tasks on datasets such as 20NG, WS, and TMN.

**Table 1.** top-Purity, top-NMI, topic diversity and document classification accuracy for document clustering. The symbols, “ $\uparrow$ ” and “ $\downarrow$ ”, indicate “the lower the better” and “the higher the better”, respectively. The best result for each dataset is in **bold**. The second result for each dataset is in underline.

	top-Purity $\uparrow$			top-NMI $\uparrow$		
	WS	20NG	TMN	WS	20NG	TMN
NSTM	0.451 $\pm$ 0.009	0.184 $\pm$ 0.011	0.554 $\pm$ 0.010	0.201 $\pm$ 0.004	0.170 $\pm$ 0.012	0.267 $\pm$ 0.004
NSTM-exp	0.295 $\pm$ 0.012	0.170 $\pm$ 0.002	0.325 $\pm$ 0.006	0.100 $\pm$ 0.008	0.120 $\pm$ 0.004	0.085 $\pm$ 0.003
NSTM-constrained	0.380 $\pm$ 0.002	0.176 $\pm$ 0.004	0.500 $\pm$ 0.015	0.197 $\pm$ 0.002	0.161 $\pm$ 0.003	0.195 $\pm$ 0.007
SPTM-TM	<b>0.454<math>\pm</math>0.007</b>	<b>0.190<math>\pm</math>0.042</b>	<b>0.555<math>\pm</math>0.028</b>	<b>0.268<math>\pm</math>0.038</b>	<b>0.176<math>\pm</math>0.020</b>	<b>0.270<math>\pm</math>0.004</b>
	topic diversity $\uparrow$			doc classification acc $\uparrow$		
	20NG	WS	TMN	20NG	WS	TMN
NSTM	0.760 $\pm$ 0.081	<b>0.911<math>\pm</math>0.013</b>	0.647 $\pm$ 0.004	<u>0.383<math>\pm</math>0.002</u>	<b>0.794<math>\pm</math>0.013</b>	<u>0.648<math>\pm</math>0.007</u>
NSTM-exp	<b>0.822<math>\pm</math>0.007</b>	0.860 $\pm$ 0.003	<b>0.835<math>\pm</math>0.004</b>	0.137 $\pm$ 0.003	0.204 $\pm$ 0.005	0.207 $\pm$ 0.006
NSTM-constrained	<u>0.797<math>\pm</math>0.002</u>	0.845 $\pm$ 0.003	0.633 $\pm$ 0.002	0.365 $\pm$ 0.003	0.716 $\pm$ 0.003	0.593 $\pm$ 0.002
SPTM-TM	0.763 $\pm$ 0.062	<u>0.910<math>\pm</math>0.011</u>	<u>0.648<math>\pm</math>0.002</u>	<b>0.395<math>\pm</math>0.003</b>	<u>0.789<math>\pm</math>0.010</u>	<b>0.668<math>\pm</math>0.005</b>

Note that here we directly applied three types of mapping to the Poincaré ball on topic embeddings of NSTM. NSTM is optimized for reconstruction and diversity. However, the various mappings to the Poincaré ball are no longer optimized for reconstruction and diversity after mapping. Our experiment is designed to test whether such mappings would impair the performance of original topic model (e.g. NSTM). From the Table 1, we can see that, except for the exponential mapping causing some detriment, the impact of the Poincaré ball constraint and SPTM is negligible, with SPTM even achieving the best performance on some datasets and metrics. Therefore, we can conclude that SPTM does not diminish the quality of the original topic model.

## 5 Conclusion

In conclusion, this paper proposes the Stereographic Projection Transition Mapping (SPTM) method, bridging the gap between Euclidean and hyperbolic

embeddings for hierarchical structures. We demonstrate the limitations of relying solely on exponential mapping and present an unsupervised approach for hierarchical embeddings. Additionally, we explore the explicit definition of the exponential mapping in the neural network and introduce a method to transfer existing neural topic models to hierarchical topic models. Our experiments further validate SPTM's efficacy in image reconstruction using the hyperbolic VQ-VAE model. Overall, SPTM offers a promising solution for efficient and effective hierarchical embeddings in hyperbolic space with potential applications in various domains.

We hope that the idea of directly mining hierarchical structures from deep neural networks through analytical methods can provide some inspiration. Differential geometry, being a mature field, should find more refined applications in deep learning.

## References

1. Bécigneul, G., Ganea, O.E.: Riemannian adaptive optimization methods. arXiv preprint [arXiv:1810.00760](https://arxiv.org/abs/1810.00760) (2018)
2. Beltrami, E.: Teoria fondamentale degli spazii di curvatura costante memoria. F. Zanetti (1868)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
4. Chen, J., Qin, J., Shen, Y., Liu, L., Zhu, F., Shao, L.: Learning attentive and hierarchical representations for 3d shape recognition. In: European Conference on Computer Vision, pp. 105–122 (2020)
5. Dhingra, B., Shallue, C.J., Norouzi, M., Dai, A.M., Dahl, G.E.: Embedding text in hyperbolic spaces. arXiv preprint [arXiv:1806.04313](https://arxiv.org/abs/1806.04313) (2018)
6. Dieng, A.B., Ruiz, F.J., Blei, D.M.: Topic modeling in embedding spaces. *Trans. Assoc. Comput. Linguist.* **8**, 439–453 (2020)
7. Doe, J., Smith, J.: Webis-snippet. <https://webis.de/data-old/webis-snippet-20.html> (2022)
8. Fang, P., Harandi, M., Petersson, L.: Kernel methods in hyperbolic spaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10665–10674, October 2021
9. Fellbaum, C.: Wordnet. In: Theory and applications of ontology: computer applications, pp. 231–243. Springer (2010)
10. Ganea, O., Bécigneul, G., Hofmann, T.: Hyperbolic neural networks. *Advances in neural information processing systems* **31** (2018)
11. Guo, Y., Wang, X., Chen, Y., Yu, S.X.: Clipped hyperbolic classifiers are super-hyperbolic classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11–20, June 2022
12. Hofmann, T.: Probabilistic latent semantic analysis. arXiv preprint [arXiv:1301.6705](https://arxiv.org/abs/1301.6705) (2013)
13. Iversen, B.: Hyperbolic geometry. No. 25. Cambridge University Press (1992)
14. Khrukov, V., Mirvakhabova, L., Ustinova, E., Oseledets, I., Lempitsky, V.: Hyperbolic image embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2020

15. Lang, K., Rainbow: The 20 newsgroups data set (2008). <http://qwone.com/~jason/20Newsgroups/>
16. Ma, R., Fang, P., Drummond, T., Harandi, M.: Adaptive poincaré point to set distance for few-shot classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 1926–1934 (2022)
17. Miao, Y., Grefenstette, E., Blunsom, P.: Discovering discrete latent topics with neural variational inference. In: International Conference on Machine Learning, pp. 2410–2419. PMLR (2017)
18. Monath, N., Zaheer, M., Silva, D., McCallum, A., Ahmed, A.: Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 714–722 (2019)
19. Murtagh, F., Contreras, P.: Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Rev. Data Mining Knowl. Discov.* **2**(1), 86–97 (2012)
20. Nagano, Y., Yamaguchi, S., Fujita, Y., Koyama, M.: A wrapped normal distribution on hyperbolic space for gradient-based learning. In: International Conference on Machine Learning, pp. 4693–4702. PMLR (2019)
21. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. *Trans. Assoc. Comput. Linguist.* **3**, 299–313 (2015)
22. Nielsen, F., Nielsen, F.: Hierarchical clustering. *Introduction to HPC with MPI for Data Science*, pp. 195–211 (2016)
23. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
24. Sala, F., De Sa, C., Gu, A., Ré, C.: Representation tradeoffs for hyperbolic embeddings. In: International Conference on Machine Learning, pp. 4460–4469. PMLR (2018)
25. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. arXiv preprint [arXiv:1703.01488](https://arxiv.org/abs/1703.01488) (2017)
26. Valentino, M., Carvalho, D.S., Freitas, A.: Multi-relational hyperbolic word embeddings from natural language definitions. arXiv preprint [arXiv:2305.07303](https://arxiv.org/abs/2305.07303) (2023)
27. Xu, Y., Wang, D., Chen, B., Lu, R., Duan, Z., Zhou, M., et al.: Hyperminer: topic taxonomy mining with hyperbolic embedding. *Adv. Neural. Inf. Process. Syst.* **35**, 31557–31570 (2022)
28. Zhang, Y., Wang, X., Shi, C., Jiang, X., Ye, Y.: Hyperbolic graph attention network. *IEEE Trans. Big Data* **8**(6), 1690–1701 (2021)
29. Zhao, H., Phung, D., Huynh, V., Le, T., Buntine, W.: Neural topic model via optimal transport. arXiv preprint [arXiv:2008.13537](https://arxiv.org/abs/2008.13537) (2020)



# SPCSE: Soft Positive Enhanced Contrastive Learning for Sentence Embeddings

Lingen Liu, Zixin Chen, and Guang Chen<sup>(✉)</sup>

Institute of Artificial Intelligence, Beijing University of Posts and  
Telecommunications, Beijing, China  
{llg, mailboxforvicky, chenguang}@bupt.edu.cn

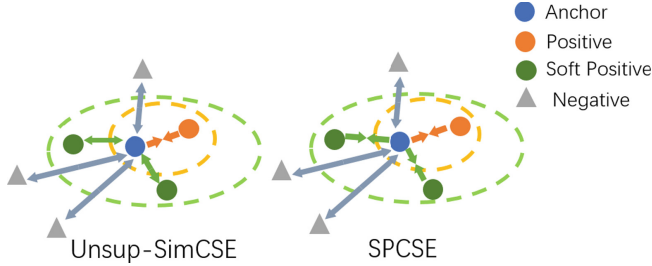
**Abstract.** Unsupervised contrastive learning for high-quality sentence representations has gained widespread attention in recent years. However, existing dropout-based data augmentation method, such as Unsup-SimCSE [13], may suffer from the limitation of minimal semantic changes, which can result in the potential exclusion of positive samples and thus hinder alignment. To alleviate this problem, we propose a novel approach called Soft Positive Contrastive Sentence Embeddings (SPCSE), which leverages soft positives generated from diverse discrete data augmentation methods. By incorporating soft positives, SPCSE aims to enhance the alignment between positive samples and anchors in the representation space. Our experimental results across seven Semantic Textual Similarity (STS) tasks demonstrate that SPCSE can significantly improve the alignment of positive samples and achieve overall performance enhancement compared to Unsup-SimCSE.

**Keywords:** Sentence representation learning · Unsupervised learning · Soft positives

## 1 Introduction

Sentence representation learning [15,17] aims to learn a universal sentence embedding that can benefit diverse downstream tasks including information retrieval and text classification. Recent studies [13,20] has demonstrated that contrastive learning can help pre-trained language models learn high-quality embeddings in an unsupervised fashion. Contrastive learning aims to learn effective sentence embedding representations by pulling positive samples closer and pushing negative samples away. Positive samples are typically derived from various data augmentations of the same instance, whereas other instances function as negative samples.

Among recent works [12,13], SimCSE [13] has emerged as a strong baseline due to its simplicity and effectiveness. Unsup-SimCSE [13] utilizes different dropout masks as minimal data augmentation, which yields remarkably superior performance, even on par with previous supervised approaches. However,



**Fig. 1.** An illustration of the induced embedding distribution. Unsup-SimCSE may unintentionally push away potential positive samples, while SPCSE helps all positives aligned by introducing soft positives.

we argue that Unsup-SimCSE is prone to the limitation of minimal semantic changes. Although dropout-based method maintains semantic consistency, it inadvertently pushes away other potential positive samples, thereby hindering the alignment of positive samples in the representation space. On the other hand, though the samples obtained by discrete data augmentation suffer from semantic deviation and underperforms dropout-based method [8, 20], they can still be viewed as positive samples, to which we refer as soft positive samples.

As shown in Table 1, we measured the semantic similarity score between the anchors and positives, soft positives and negatives. Although the similarity between soft positives and anchors is lower than that between positives and anchors, from both theoretical perspective and the semantic similarity score, soft positives can still be considered a supplement to positive samples.

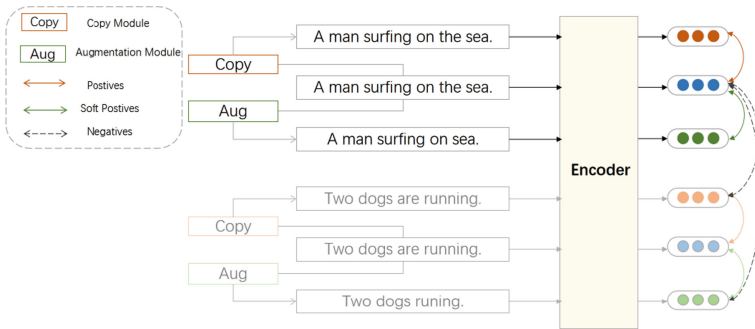
**Table 1.** We performed semantic relevance statistics between the anchor samples and positive samples, soft positive samples, and negative samples. We used the unsupervised SimCSE as the encoder and calculated the cosine similarity as the semantic relevance score.

Sample Pair	Semantic Similarity Score
anchor and positives	$0.95 \pm 0.04$
anchor and soft positives	$0.90 \pm 0.09$
anchor and negatives	$0.57 \pm 0.15$

Therefore, we propose utilizing soft positive samples obtained by discrete data augmentation to enhance the alignment of positive samples in the representation space.

By introducing soft positive samples, we can not only significantly expand the set of positive samples but also enhance the alignment capability of the embedding representations through contrastive learning.

Our primary goal is to enhance the alignment performance of positive samples by introducing soft positives. Specifically, we obtain soft positives through discrete data augmentation and introduce additional contrastive learning between anchor samples and soft positives. Therefore, SPCSE can facilitate a more clustered distribution of positive samples, and improve the performance of embeddings. As shown in Fig. 1, Unsup-SimCSE may push away potential positive samples, resulting in poor alignment performance. In contrast, SPCSE introduces additional soft positives during training, allowing the model to better focus on positive samples, thereby enhancing the alignment of positive samples (Fig. 2).



**Fig. 2.** The overview of SPCSE framework. For a input sentence, we get its positive samples and soft positive samples by different dropout mask and discrete data augmentation with other in-batch samples as negatives. For output embeddings, We do contrastive learning between anchor and positives and soft positives respectively, in order to improve the alignment between positive samples.

We evaluate our approach on seven Semantic Textual Similarity tasks. Experimental results demonstrate that SPCSE outperforms Unsup-SimCSE by an average Spearman correlation of 1.96% and 1.25% on  $BERT_{base}$  and  $BERT_{large}$  models, respectively. To show the effectiveness of SPCSE, we measure its alignment and uniformity [18] performance on the STS-B development set. The results indicate that SPCSE significantly improves the alignment performance, achieving relative improvements of 13.4% and 16.9% on  $BERT_{base}$  and  $BERT_{large}$ . However, at the same time, the clustering of soft positives results in a slight decline in uniformity performance. Our subsequent experimental findings demonstrate the crucial role of temperature in achieving a balance between these two essential attributes. Excessive high or low temperatures will lead to a deterioration in model performance.

Our contributions can be summarized as follows: We propose a novel unsupervised contrastive learning framework to enhance the alignment of representation, by introducing soft positive generated from discrete data augmentation. Our experimental results demonstrate that SPCSE can effectively improve the

alignment of positive samples, and improve its performance on STS tasks. We have also released our code for further study<sup>1</sup>.

## 2 Related Works

### 2.1 Contrastive Learning in Sentence Representation

Recently, contrastive learning for sentence representation has achieved significant success. Yan [20] firstly incorporates multiple data augmentation strategies, such as token shuffling and cutoff. SimCSE [13] employs different dropout masks to get positive pairs, demonstrating the effectiveness of this straightforward approach compared to other data augmentation strategies. [14] propose a prompt-based sentence embeddings method and utilize BERT layers more effectively. Current methods mainly focus on using different data augmentation strategies to generate better positive pairs [10,12]. Chuang [8] introduces an additional ELECTRA [28] model as a discriminator to differentiate the representation of the encoder, while trans-Encoder [27] uses dropout noise to train the encoder, they focused on the distillation stage in the subsequent training steps. However, they neglected the alignment between positive samples that have lower semantic similarity and anchor samples, which holds equal importance for unsupervised sentence representation.

### 2.2 Positive and Negative Instances

One critical question in unsupervised contrastive learning is how to construct the triplet  $(x_i, x_i^+, x^-)$ . In visual representation learning, an effective solution is to take two different transformations of the same image (e.g., cropping, flipping, distortion and rotation) as  $x_i$  and  $x_i^+$ , where negatives  $x_i^-$  are typically randomly sampled from the same batch. However, in NLP, the way to obtain positive and negative sample triplets differs. Based on the context of samples, we can categorize these methods into context-based and transformation-based approaches.

In the context-based approach, positive samples typically come from the same document or paragraph, while negative samples are randomly sampled from other documents. In the transformation-based approach, positive samples can be obtained through discrete data augmentation (e.g., random delete, random shuffle, etc.) or continuous data augmentation(dropout mask). SimCSE has demonstrated that continuous data augmentation is more effective than discrete data augmentation with better model performance. In this paper, we introduce SP samples obtained through discrete data augmentation to mitigate the limitations of the dropout method.

---

<sup>1</sup> <https://github.com/ilingen/SPCSE>.



### 2.3 Alignment and Uniformity in Contrastive Learning

Recently, Wang [18] proposed two important attributes of contrastive learning: alignment and uniformity, and utilized them to measure the quality of representations. Given a distribution of positive samples  $p_{pos}$ , alignment measures the closeness of positive pairs, while uniformity measures how well the embeddings are uniformly distributed.

$$\mathcal{L}_{\text{align}} \triangleq \mathbb{E}_{(x, x^+) \sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2,$$

$$\mathcal{L}_{\text{uniform}} \triangleq \log \mathbb{E}_{\substack{i.i.d. \\ x, y \sim p_{\text{data}}}} e^{-2\|f(x) - f(y)\|^2}$$

These two metrics are well aligned with the objective of contrastive learning: positive instances should stay close and embeddings for random instances should scatter on the hypersphere. Gao [13] used them to evaluate their models and found that optimizing both of these properties can improve the quality of embeddings. In the following sections, we will also analyze our approach on these two properties in Sect. 5, and show how our approach works.

## 3 Approach

### 3.1 Unsupervised Contrastive Learning

Contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors Hadsell [1]. Given a set of paired examples  $D = \{(x_i, x_i^+)\}_{i=1}^m$ , where  $x_i$  and  $x_i^+$  are semantically related. Let  $e_i$  denote the representation vector of  $x_i$ . For a batch with  $N$  pairs, we first conduct contrastive learning between anchor and positive instance. Training objective  $L_{CL}$  for  $(e_i, e_i^+)$  within a batch of  $N$  pairs is:

$$L_{CL} = \sum_{i=0}^N -\log \frac{\exp(\text{sim}(e_i, e_i^+)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(e_i, e_j^+)/\tau)} \quad (3.1)$$

where  $\tau$  is a temperature hyper-parameter and  $\text{sim}(e_1, e_2)$  is the cosine similarity  $\frac{e_1^T e_2}{\|e_1\| \|e_2\|}$ . In unsupervised contrastive learning, positive pairs are typically obtained by applying different data augmentation methods to the same sample. In this work, we encode input sentences using a pre-trained language model such as BERT [11]:  $e = f_\theta(x)$ , and then fine-tune all the parameters using the contrastive learning objective (Eq. 3.1).

### 3.2 Contrastive Learning with Soft Positive

Our approach incorporates two contrastive learning loss functions: one between the anchor and positive samples, and the other between the anchor and soft positive samples. Specifically, given a batch of sentence pairs:  $D = \{(x_i, x_i^+, x_{i_{sp}}^+)\}_{i=1}^N$ ,

where  $x_i$  and  $x_i^+$  are the  $i$ th positive pair,  $x_i$  and  $x_{i_{sp}}^+$  are the  $i$ th soft positive pair. Let  $e_i$  denote the representation vector of  $x_i$ . For a batch with  $N$  pairs, we first do contrastive learning between the anchor and positive instances. Training objective for  $(e_i, e_i^+)$  within a batch of  $N$  pairs is  $L_{CL}$  (Eq. 3.1).

We also perform contrastive learning between the anchor and soft positive samples, the training objective  $L_{CL_{sp}}$  is:

$$L_{CL_{sp}} = \sum_{i=0}^N -\log \frac{\exp(\text{sim}(e_i, e_{i_{sp}}^+)/\tau_{sp})}{\sum_{j=1}^N \exp(\text{sim}(e_i, e_{j_{sp}}^+)/\tau_{sp})} \quad (3.2)$$

In our approach, we use the same encoder for the encoded output, the difference is only in the input sample pair (i.e.  $(e_i, e_i^+)$  in  $L_{CL}$  and  $(e_i, e_{i_{sp}}^+)$  in  $L_{CL_{sp}}$ ) and the temperature hyper-parameter.

Meanwhile, As mentioned in the introduction, soft positive samples come from discrete data augmentation and positive samples from dropout noise. Therefore, ideally, the semantic similarity score between anchor and positive should be greater than the score between anchor and soft positive (i.e.  $\text{sim}(e_i, e_i^+) \geq \text{sim}(e_i, e_{i_{sp}}^+)$ ). To keep the relative order among the triplets, we added an additional regularization loss objective (Eq. 3.3).

$$L_{reg} = \sum_{i=0}^N \max(\text{sim}(e_i, e_{i_{sp}}^+) - \text{sim}(e_i, e_i^+), m) \quad (3.3)$$

In which  $\text{sim}(e_i, e_j)$  is the cosine similarity between  $e_i$  and  $e_j$ ,  $m$  is the margin for regularization loss function, Here we set the margin  $m = 0$ . Finally, the overall loss is (Eq. 3.4):

$$L = L_{CL} + \lambda_1 L_{CL_{sp}} + \lambda_2 L_{reg} \quad (3.4)$$

Here  $\lambda_1$  and  $\lambda_2$  are coefficient parameters.

## 4 Experiments

### 4.1 Evaluation Tasks

Following previous works, we conduct our experiments on seven standard STS tasks. For all these tasks, we use the SentEval toolkit [9] for evaluation. We evaluate our approach on Semantic Textual Similarity(STS) tasks: STS 2012-2016 [2–6], STS Benchmark [7] and SICK-Relatedness [16]. We use Spearman’s Correlation coefficient as performance metric.

### 4.2 Training Details

We use Unsup-SimCSE as our baseline model. To simplify the training process, We use EDA [19] as our data augmentation tool, which includes four augmentation methods: synonym replacement (SR), random insertion (RI), random swap

(RS), and random deletion (RD). The cropping ratio is set at 30%. More details about EDA tools can be found in Appendix A.2.

We use BERT<sub>base</sub> and BERT<sub>large</sub> [11] as our base model. During training, for a given sentence in the training set, we randomly choose and perform one of the above operations to generate soft positive. As for hyper-parameters, We set  $\tau=0.05, \tau_{sp}=0.5, \lambda_1=1e-4, \lambda_2=1e-4$ , while other parameters remain consistent with SimCSE. More training details can be found in Appendix A.1.

### 4.3 Main Results

As depicted in Table 2, SPCSE demonstrates superior performance over SimCSE with improvements of 1.97% and 1.04% on seven STS tasks using both BERT<sub>base</sub> and BERT<sub>large</sub> models, thus demonstrating the efficacy of our proposed approach. Compared with other concurrent works [20, 23, 25], SPCSE has also achieved excellent performance. In most cases, SPCSE can demonstrate comparable performance to ESimCSE [24].

**Table 2.** Sentence embedding performance on STS test sets in terms of Spearman’s correlation.  $\heartsuit$ : results are reproduced and reevaluated by [13].  $\diamond$ : results from [20].  $\clubsuit$ : results from [25].  $\spadesuit$ : results from [13].  $\triangle$ : results from [24].  $\blacklozenge$ : results from [23].  $\blacktriangleleft$ : results from [8].  $\blacktriangleright$ : results from [27].

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
CT-BERT <sub>base</sub> <sup><math>\heartsuit</math></sup>	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
ConSERT-BERT <sub>base</sub> <sup><math>\diamond</math></sup>	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SG-OPT-BERT <sub>base</sub> <sup><math>\clubsuit</math></sup>	66.84	80.13	71.23	81.56	77.17	77.23	68.23	74.62
SimCSE-BERT <sub>base</sub> <sup><math>\spadesuit</math></sup>	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
SDA-BERT <sub>base</sub> <sup><math>\blacklozenge</math></sup>	71.84	83.79	75.49	82.91	78.55	78.73	70.12	77.35
ESimCSE-BERT <sub>base</sub> <sup><math>\triangle</math></sup>	<b>73.40</b>	83.27	<b>77.25</b>	82.66	78.81	80.17	72.30	78.27
DiffCSE-BERT <sub>base</sub> <sup><math>\blacktriangleleft</math></sup>	72.28	<b>84.43</b>	76.47	<b>83.90</b>	80.54	80.59	71.23	78.49
Trans-Encoder-BERT-bi <sub>base</sub> <sup><math>\blacktriangleright</math></sup>	72.17	84.40	76.69	83.28	<b>80.91</b>	<b>81.26</b>	71.84	<b>78.65</b>
Trans-Encoder-BERT-cross <sub>base</sub> <sup><math>\blacktriangleright</math></sup>	71.94	84.14	76.39	82.87	80.65	81.06	71.16	78.32
SPCSE-BERT <sub>base</sub> (Our)	72.80	83.82	76.00	83.11	79.78	80.07	<b>71.94</b>	78.13
ConSERT-BERT <sub>large</sub> <sup><math>\diamond</math></sup>	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
SG-OPT-BERT <sub>base</sub> <sup><math>\clubsuit</math></sup>	67.02	79.42	70.38	81.72	76.35	76.16	70.20	74.46
SimCSE-BERT <sub>large</sub> <sup><math>\spadesuit</math></sup>	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
ESimCSE-BERT <sub>large</sub> <sup><math>\triangle</math></sup>	73.21	85.37	<b>77.73</b>	84.30	78.92	80.73	74.89	79.31
Trans-Encoder-BERT-bi <sub>large</sub> <sup><math>\blacktriangleright</math></sup>	75.55	84.08	77.01	85.43	<b>81.37</b>	82.88	71.46	79.68
Trans-Encoder-BERT-cross <sub>large</sub> <sup><math>\blacktriangleright</math></sup>	<b>75.81</b>	84.51	76.50	<b>85.65</b>	82.14	<b>83.47</b>	70.90	<b>79.85</b>
SPCSE-BERT <sub>large</sub> (Our)	73.85	<b>85.83</b>	77.68	85.05	79.17	80.79	<b>75.04</b>	79.63

#### 4.4 Ablation Study

**Effect of Data Augmentations.** In our work, we believe that SimCSE [13] is limited in using dropout as data augmentation tools, and thus it will hinder the model’s ability to gather potential soft positive samples. To simplify experimental process, we use data enhancement methods in EDA [19] (random delete (RD), random swap (RS), random insert (RI), synonym replace (SR)) to construct soft positive samples. We introduce anchor samples and soft positive samples and keep the distance between soft positive samples and anchor samples. Based on BERT<sub>base</sub>, we evaluated the experimental results of different type of data augmentations in Table 3. As can be seen from the Table 3, different data augmentation methods have different effects on the performance of the model, with synonym substitution bringing the most benefit. SPCSE uses the combination of four methods to achieve the best performance, which indicates that the combination of multiple enhancement methods is also conducive to the improvement of model performance.

**The Importance of Proposed Additive Objective.** To demonstrate the effectiveness of our proposed additional method, we added an ablation study experiment. In this section, we separately removed the contrastive learning loss functions of the anchor samples and soft positive samples, as well as the relative relationship constraint loss function, from the proposed SPCSE framework. We then measured the average score on the STS dataset.

**Table 3.** The Effect of different augmentation on BERT<sub>base</sub>. We evaluate and report its result on STS task.

Type of augmentations	STS Avg
SimCSE	76.14
SimCSE+RD	77.04
SimCSE+RS	77.31
SimCSE+RI	76.83
SimCSE+SR	77.58
SPCSE	78.13

**Table 4.** The Effect of different augmentation on BERT<sub>base</sub>. We evaluate and report its result on STS task.

loss function	STS Avg
SPCSE w/o $L_{CL_{sp}}$ and $L_{reg}$	76.14
SPCSE w/0 $L_{CL_{sp}}$	76.94
SPCSE w/0 $L_{reg}$	77.68
SPCSE	78.13

As shown in Table 4, the absence of either the contrastive loss for soft positive and anchor samples or the relative relationship constraint function led to a significant decline in model performance. Including only the relative relationship constraint function can effectively improve model performance, but it is clearly less effective than the improvement brought by the contrastive learning of soft positive samples. This also confirms our hypothesis that soft positive samples can be considered a supplement to positive samples. Additionally, it is crucial not

to treat all samples obtained through data augmentation as positive samples indiscriminately; the relative magnitude relationships among positive samples must be maintained.

## 5 Analysis

In this section, we further study the effectiveness of our proposed SPCSE.

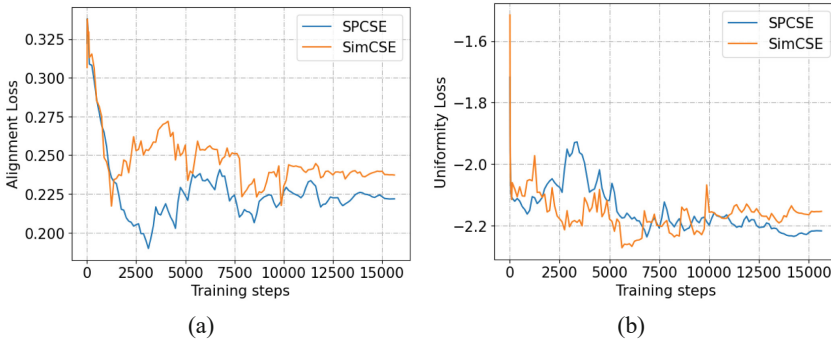
### 5.1 Alignment-Uniformity Analysis

As shown in Sect. 2.3, alignment and uniformity are two key properties to measure the quality of embeddings. To validate the improvement of the alignment of SPCSE, we compare the alignment and uniformity loss of SPCSE and SimCSE using  $BERT_{base}$  during training, and also measure SPCSE and SimCSE’s final alignment and uniformity results.

As shown in Fig. 3 and Table 5, we can see that SPCSE significantly improves alignment performance with only a slight drop in uniformity. This can be

**Table 5.** SimCSE and SPCSE’s performance on Alignment-Uniformity under 5 different random seeds. SPCSE can improve the model’s alignment performance while maintaining a minimal decrease in uniformity.

Model	Alignment	Uniformity
SimCSE- $BERT_{base}$	0.238±0.012	<b>-2.248±0.139</b>
SPCSE- $BERT_{base}$	<b>0.204±0.011</b>	-2.104±0.148
SimCSE- $BERT_{large}$	0.236±0.020	<b>-2.290±0.228</b>
SPCSE- $BERT_{large}$	<b>0.196±0.016</b>	-2.224±0.127



**Fig. 3.** The Alignment and Uniformity loss of SPCSE and Unsup-SimCSE using  $BERT_{base}$  on the validation set of STS-B during training. For these two attributes, smaller values indicate a better distribution. SPCSE can significantly improve alignment performance with slight drop in uniformity.

attributed to the incorporation of soft positive in SPCSE, which enhances the alignment of sentence representations. Meanwhile, due to the further aggregation of positive samples, the uniformity decreases. We will discuss how to balance these two key properties in Sect. 5.2.

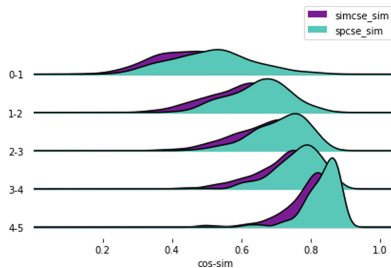
## 5.2 Hyper-parameters Analysis

**The Role of  $\tau_{sp}$ .** Prior work [13,18] found that  $\tau$  can balance alignment and uniformity, thereby controlling the quality of the representation. To demonstrate the effectiveness of  $\tau_{sp}$  in SPCSE, we measure the performance of SPCSE with different  $\tau_{sp}$  on the STS dataset, as well as the corresponding alignment and uniformity values. Results were visualized in Fig. 5(a). The role of the  $\tau_{sp}$  can be concluded as follows: 1). Lower  $\tau_{sp}$  improves alignment performance, but will harm the uniformity a lot. 2). While a higher  $\tau_{sp}$  ensures stability in uniformity, it can result in insufficient alignment of positives. 3). In our experiment, excessive pursuit of alignment can result in model degradation, we should keep the  $\tau_{sp}$  within a reasonable interval to avoid this situation.

**Coefficient  $\lambda_1$  and  $\lambda_2$ .** For hyper-parameters analysis, we study the impact of  $\lambda_1$  and  $\lambda_2$ . Both of these parameters are used to balance the magnitudes of the loss functions, so they need to be selected within an appropriate interval. According to Sect. 5.2, we set  $\tau_{sp}=0.5$  and evaluate SPCSE with varying values  $\lambda_1$  and  $\lambda_2$  on the STS-B tasks using the BERT<sub>base</sub> model. Figure 5(b) shows the influence of the  $\lambda_1$  and  $\lambda_2$  on the STS-B tasks. For both  $\lambda_1$  and  $\lambda_2$ , too large or too small values may lead to a performance degradation. The reason may be that inappropriate values will lead to an imbalance of loss magnitudes. We should choose an appropriate value for  $\lambda_1$  and  $\lambda_2$  during training.

## 5.3 Distribution of Sentence Embedding

To show the representation space of SPCSE, we plot the cosine similarity distribution of sentence pairs from STS-B test set for both SimCSE and SPCSE in



**Fig. 4.** The distribution of cosine similarities from SimCSE/DiffCSE for STS-B test set. Along the y-axis are 5 groups of data splits based on human ratings. The x-axis is the cosine similarity.

Fig. 4. We can observe that both SimCSE and SPCSE can assign cosine similarities consistent with human ratings. The SPCSE model exhibits a more compressed and hierarchically structured cosine similarity distribution compared to SimCSE. This further substantiates our proposition that the introduction of soft positives can effectively enhance the alignment performance of samples. However, we also find that under the same human rating, SPCSE assigns slightly higher cosine similarities compared with SimCSE. This phenomenon aligns with our expectations. While the incorporation of soft positive samples in contrastive learning can bolster the alignment among samples, it concurrently compromises the uniformity of the sample space, as demonstrated in Sect. 5.1. Consequently, although the presence of soft positive samples can enhance the alignment performance of contrastive learning models, there still remains a requisite trade-off between the alignment and uniformity within the sample space. As explicated in Sect. 5.2, an appropriate temperature coefficient can effectively balance these two factors.

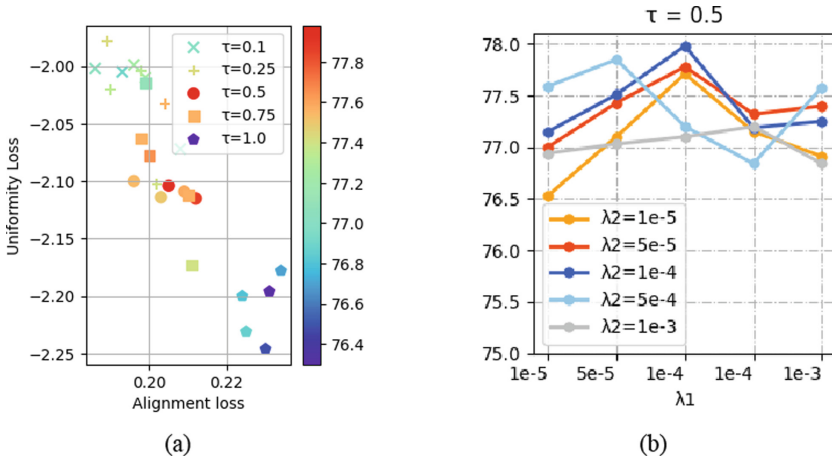


Fig. 5. The impact of hyper-parameters on model performance: (a) represents the effects of different  $\tau_{sp}$  on alignment-uniformity and STS(Avg) results, while (b) represents the effects of  $\lambda_1$  and  $\lambda_2$  on STS(Avg) results.

## 6 Conclusion

In this paper, we propose SPCSE, a soft positive enhanced contrastive learning framework for unsupervised sentence representation learning. Our main goal is to improve the alignment performance of positive samples with soft positives.

To achieve this, we employ discrete augmentation to generate soft positive samples and perform two types of contrastive learning separately.

Our experiments on alignment and uniformity demonstrate that SPCSE can significantly improve the alignment of positive samples. Experimental results on seven STS tasks have shown that our approach outperforms competitive baseline.

In the future, we will explore how to improve the generalization capability of SPCSE and verify its effectiveness on other contrastive learning methods.

## A Appendix A

### A.1 Training Detail

We use SimCSE as our baseline model. Training data contains one million sentences crawled from Wikipedia. For positives obtained by dropout mask, we extract sentence embedding using a fine-tuned BERT model and use two independent dropout masks. For soft positives, we firstly generate four augmented sentences for each sample with EDA tools. During training, one of the four augmented sentences is randomly chosen as soft positive sample. The cropping ratio is 30%. The hyper-parameters settings are listed in Table 6

**Table 6.** Training details of SPCSE.

Model	$\lambda_1$	$\lambda_2$	$\tau$	$\tau_{sp}$	m
BERT <sub>base</sub>	1e-4	1e-4	0.05	0.5	0.0
BERT <sub>large</sub>	1e-4	5e-4	0.05	0.75	0.0

We use the BERT<sub>base</sub> and BERT<sub>large</sub> models with respective learning rates  $3e-5$  and  $1e-5$ . We train both models for one epoch with batch size 64. We use early stopping to avoid overfitting. Our code is implemented in Python 3.6, using Pytorch 1.60, and the experiments are run on a single 48G NVIDIA A6000 GPU.

### A.2 Discrete Data Argumentation Methods

We used four types of discrete data augmentation from EDA as the source of soft positives. Here, we will show the full details of EDA [19]. For a given sentence in the training set, we randomly choose and perform one of the following operations, As shown in Table 7:

**1. Synonym Replacement (SR):** Randomly choose  $n$  words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.

**2. Random Insertion (RI):** Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this  $n$  times.



**3. Random Swap (RS):** Randomly choose two words in the sentence and swap their positions. Do this  $n$  times.

**4. Random Deletion (RD):** Randomly remove each word in the sentence with probability  $p$ .

**Table 7.** Sentences generated using EDA.

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
Synonym Replacement	A <b>lamentable</b> , superior human comedy played out on the <b>backward</b> road of life.
Random Insertion	A sad, superior human comedy played out on <b>funniness</b> the back roads of life.
Random Swap	A sad, superior human comedy played out on <b>roads</b> back <b>the</b> of life.
Random Deletion	A sad, superior human out on roads back the of life.

## References

1. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, pp. 1735–1742 (2006)
2. Agirre, E., et al.: Semeval-2014 task 10: multilingual semantic textual similarity. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), pp. 81–91 (2014)
3. Agirre, E., et al.: Semeval-2015 task 2: semantic textual similarity, English, Spanish and pilot on interpretability. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 252–263 (2015)
4. Agirre, E., et al.: Semeval-2016 task 1: semantic textual similarity, monolingual and cross-lingual evaluation. In: SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511. ACL (Association for Computational Linguistics) (2016)
5. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: \* sem 2013 shared task: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (\* SEM), volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 32–43 (2013)
6. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A.: Semeval-2012 task 6: a pilot on semantic textual similarity. In: \* SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pp. 385–393 (2012)
7. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint [arXiv:1708.00055](https://arxiv.org/abs/1708.00055) (2017)
8. Chuang, Y.-S., et al.: Diffcse: difference-based contrastive learning for sentence embeddings. arXiv preprint [arXiv:2204.10298](https://arxiv.org/abs/2204.10298) (2022)

9. Conneau, A., Kiela, D.: Senteval: an evaluation toolkit for universal sentence representations. arXiv preprint [arXiv:1803.05449](https://arxiv.org/abs/1803.05449) (2018)
10. Cao, R., Wang, Y., Liang, Y., Gao, L., Zheng, J., Ren, J., Wang, Z.: Exploring the impact of negative samples of contrastive learning: a case study of sentence embeddin. arXiv preprint [arXiv:2202.13093](https://arxiv.org/abs/2202.13093) (2022)
11. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
12. Giorgi, J., Nitski, O., Wang, B., Bader, G.: Declutr: deep contrastive learning for unsupervised textual representations. arXiv preprint [arXiv:2006.03659](https://arxiv.org/abs/2006.03659) (2020)
13. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Empirical Methods in Natural Language Processing (EMNLP) (2021)
14. Jiang, T., et al.: Promptbert: improving bert sentence embeddings with prompts. arXiv preprint [arXiv:2201.04337](https://arxiv.org/abs/2201.04337) (2022)
15. Logeswaran, L., Lee, H.: An efficient framework for learning sentence representations. arXiv preprint [arXiv:1803.02893](https://arxiv.org/abs/1803.02893) (2018)
16. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., Zamparelli, R., et al.: A sick cure for the evaluation of compositional distributional semantic models. In: Lrec, pp. 216–223. Reykjavik (2014)
17. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. arXiv preprint [arXiv:1908.10084](https://arxiv.org/abs/1908.10084) (2019)
18. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2495–2504 (2021)
19. Wei, J., Zou, K.: Eda: easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint [arXiv:1901.11196](https://arxiv.org/abs/1901.11196) (2019)
20. Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., Xu, W.: ConSERT: a contrastive framework for self-supervised sentence representation transfer. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), August 2021
21. Zhang, Y., Zhang, R., Mensah, S., Liu, X., Mao, Y.: Unsupervised sentence representation via contrastive learning with mixing negatives. In: Proceedings of the AAAI Conference on Artificial Intelligence 36, pp. 11730–11738 (2022)
22. Zhou, K., Zhang, B., Zhao, W.X., Wen, J.-R.: Debaised contrastive learning of unsupervised sentence representations. arXiv preprint [arXiv:2205.00656](https://arxiv.org/abs/2205.00656) (2022)
23. Mao, Z., Zhu, D., Lu, J., Zhao, R., Tan, F.: Sda: simple discrete augmentation for contrastive sentence representation learning. arXiv preprint [arXiv:2210.03963](https://arxiv.org/abs/2210.03963) (2022)
24. Wu, X., Gao, C., Zang, L., Han, J., Wang, Z., Hu, S.: Esimcse: enhanced sample building method for contrastive learning of unsupervised sentence embedding. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 3898–3907 (2022)
25. Kim, T., Yoo, K.M., Lee, S.: Self-guided contrastive learning for bert sentence representations. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2528–2540 (2021)
26. Wang, H., Li, Y., Huang, Z., Dou, Y., Kong, L., Shao, J.: Sncse: contrastive learning for unsupervised sentence embedding with soft negative samples. arXiv preprint [arXiv:2201.05979](https://arxiv.org/abs/2201.05979) (2022)

27. Liu, F., Jiao, Y., Massiah, J., Yilmaz, E., Havrylov, S.: Unsupervised sentence-pair modelling through self- and mutual-distillations, Trans-encoder (2021)
28. Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators <https://openreview.net/pdf?id=r1xMH1BtvB> (2020)



# Neural Topic Model with Distance Awareness

Shangyu Chen<sup>1</sup>✉, He Zhao<sup>2</sup>, Viet Huynh<sup>3</sup>, Dinh Phung<sup>1</sup>,  
and Jianfei Cai<sup>1</sup>

<sup>1</sup> FIT-DSAI, Monash University, Exhibition Walk, Clayton, VIC 3168, Australia

{shangyu.chen,dinh.phung,jianfei.cai}@monash.edu

<sup>2</sup> CSIRO's Data61, Garden St, Eveleigh, NSW 2015, Australia

he.zhao@ieee.org

<sup>3</sup> School of Science, Edith Cowan University,  
270 Joondalup Dr, Joondalup, WA 6027, Australia

v.huynh@ecu.edu.au

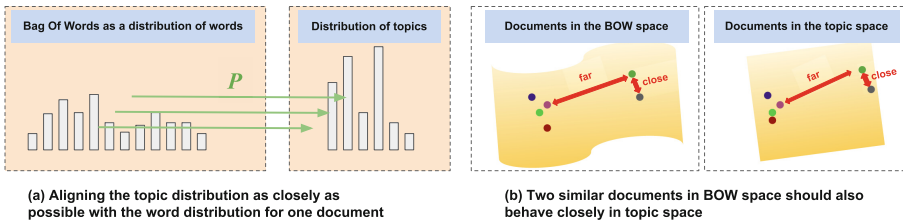
**Abstract.** Neural topic models (NTMs) have shown their success in topic modeling with a wide range of applications in text analysis. NTMs based on generative models prioritize document representations with good reconstruction capabilities, but they are insufficient in preserving distances between documents in the topic space. To bridge this gap, inspired by manifold learning, we propose a neural topic model that enables the reflection of word-to-word relationships onto topic-to-topic associations. This is achieved by approximating the distances between documents in the word space within the topic space. Extensive experiments demonstrate that the proposed model outperforms state-of-the-art NTMs in improving the quality of learned topics, as evidenced by metrics such as purity, diversity, coherence. Beyond that, the model can provide more interpretable low dimensional visualizations of documents.

**Keywords:** Manifold learning · Neural topic models

## 1 Introduction

Leveraging unsupervised techniques to extract document topics without any categories or labels is a natural idea, and topic models have become widely adopted for automated text analysis [3, 11, 30]. The topics learned through unsupervised methods can be treated as the data representation of the document. Recently, Neural Topic Models (NTMs) [13, 34, 46] leveraging Variational Autoencoders (VAE) [20] have emerged as a powerful unsupervised learning approach. Utilizing neural networks, these models efficiently organize documents into coherent topics by taking document representations as input and approximating the posterior distribution of latent topics. NTM can better overcome the problem of short text sparsity, and it also improvement over previous methods in terms of topic coherence and diversity [49].

NTM primarily aim to achieve low reconstruction error [13,34], but it cares less about the interpretability of topic representations, specifically, whether the mapping between topics and text is clearly organized or the relationships that exist between texts are preserved in the topic space. A good topic model should ensure that two semantically similar documents are geometrically close in the topic space. Recently, an NTM based on the optimal transport (OT) framework proposed in [50], named as NSTM, seeks to strike a better balance between obtaining good document representation and generating coherent and diverse topics. The model utilizes an encoder that outputs the topic distribution of the document by taking its word count vector as input like a standard NTM, but minimizes the OT distance between word count vector and topic distributions, which are two discrete distributions of the support for words and topics, as shown in Fig. 1(a). NSTM focuses on aligning the topic distribution as closely as possible with the word distribution of a document, it does not try to preserve word-to-word relationships to topic-to-topic relationships, which means it does not maintain the metric (distance relationships) of the sample space of word distributions across documents. In other words, it is natural in topic models two similar documents should be semantically similar in the topic space, however, this is not preserved in NSTM.



**Fig. 1.** Figure (a) represents the Bag of Words distribution and the topic distribution of a document. Each bar on the left represents a word, and each bar on the right represents a topic.  $P$  denotes the transport plan. In Figure (b), each point represents a document, and we aim to preserve the distance relationship between documents in both spaces.

We propose the *Distance Awareness NTM (DA-NTM)* to address this issue, by leveraging manifold learning. The main idea of manifold learning [26, 27] is to map high-dimensional data to low-dimensional data, so that the low-dimensional data can reflect some essential structural features of the original high-dimensional data. The premise of manifold learning is an assumption that some high-dimensional data is actually a low-dimensional manifold structure embedded in a high-dimensional space. A “manifold” refers to a region that is connected together, and mathematically, it refers to a set of points, each of which has its neighbors. Given any point, its manifold locally looks like Euclidean space. In other words, it has the properties of Euclidean space in the local space, and can use Euclidean space for distance calculation. Therefore, it is easy to

establish a dimension reduction mapping relationship locally, and then try to generalize the local relationship to the global, and then display it visually. From this process, it can be seen that dimension reduction is merely a byproduct of manifold learning. The primary focus is on preserving the distances between points and the neighborhoods of points while transitioning from one manifold to another.

Our DA-NTM proposes a neural topic model in this paper, which is established on a novel manifold learning framework derived from optimal transport topic modeling. This approach enables comprehensive management of the topic model training from word-to-word, topic-to-topic, and word-to-topic relationships. To briefly introduce our model, we consider two representation encodings for documents: Bag-of-word (BoW) document representations  $x$  (can be obtained by word count vector), and topic distribution  $z$ . In the document space, the vocabulary size is very large, even if a document contains only a small part of the vocabulary,  $x$  still needs to contain information about each word in the vocabulary, so  $x$  is high-dimensional and sparse. The topic distribution  $z$  is obtained from the encoder, and  $z$  can be defined as a low-dimensional vector. In NSTM [50], the learning process of the topic model is the process of the distribution of  $z$  approaching the distribution of  $x$ . In this paper, while retaining the document reconstruction ability of VAE and the distribution approximation ability of OT, we will emphasize the structural preservation ability of topic distribution to document collections, i.e., two similar documents should also behave closely in topic space, as shown in Fig. 1(b), which is achieved by manifold learning.

We summarize *our contributions* as follows: (i) We proposed a novel model DA-NTM, which combines deep topic modeling and manifold learning jointly for topic modeling of documents; (ii) The proposed model shows its benefits in both aspects of topic modeling and visualization via the comprehensive experiments; (iii) We demonstrate that DA-NTM is not limited to a specific document-to-document metric or manifold learning method, but rather represents a flexible and extensible approach.

## 2 Background

### 2.1 Neural Topic Models and Optimal Transport

Most of the existing NTMs [13, 29, 34] are neural topic models based on variational autoencoders (VAEs). The target of VAE [20] is to model the true posterior distribution  $p(z|x)$  of the latent variable, by solving the variational posterior distribution  $q_\theta(z|x)$  ( $\theta$  is a variational parameter), and continuously reduce the difference between  $q_\theta(z|x)$  and  $p(z|x)$  for approximation purpose.

The architecture of the standard VAE described above has been applied in topic modeling where each document consists of a word count (BOW, Bag Of Word) vector (i.e.  $x \in \mathbb{N}^V$ ) and a latent distribution (i.e.  $z \in \mathbb{R}^K$ ) of  $K$  topics. In order to maintain the data scale consistency, we sample from the latent space and then use them as topic distributions. An NTM assumes that the topic  $z$

of the document is determined by the prior distribution  $p(z)$  and that  $x$  can be generated by the conditional distribution  $p_\phi(x|z)$ , which is modeled by the decoder  $\phi$ . The output goal of this model is to infer the topic given a bag of words, i.e. to compute  $p(z|x)$ . Therefore, we need a neural network based the encoder  $\theta$  to get the distribution  $q_\theta(z|x)$  for approximation  $p(z|x)$ . So we have an optimization objective similar to VAE:

$$\max_{\theta, \phi} (E_{q_\theta(z|x)}[\log p_\phi(x|z)] - \mathbb{KL}[q_\theta(z|x)||p(z)]), \quad (1)$$

where the first term is the expectation of the log-likelihood, which can be understood as the reconstruction error of the document bag of words, and the second term is the Kullback-Leibler (KL) Divergence fitting the prior  $p(z)$  with  $q_\theta(z|x)$ .  $\phi(z)$  is usually constructed by a single-layer network and  $p(z)$  is usually a Gaussian distribution [41].

Next, we discuss transport(OT) [16, 31, 47, 50] for discrete distributions. The OT distance of two discrete probability distributions  $\mathbf{r}$  and  $\mathbf{c}$  can be defined as

$$d_M(\mathbf{r}, \mathbf{c}) := \min_{\mathbf{P} \in U(\mathbf{r}, \mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle, \quad (2)$$

where  $\mathbf{r} \in \Delta^{D_r}$  and  $\mathbf{c} \in \Delta^{D_c}$ , and  $\Delta^D$  represents a  $D - 1$  simplex.  $\mathbf{P} \in \mathbb{R}_{\geq 0}^{D_r \times D_c}$  is the transport plan,  $U(\mathbf{r}, \mathbf{c})$  is the transport polytope of  $\mathbf{r}$  and  $\mathbf{c}$  which means the collection of all possible transport plan.  $\mathbf{M} \in \mathbb{R}_{\geq 0}^{D_r \times D_c}$  is the cost matrix. How to define an appropriate  $\mathbf{M}$  in specific problems is a common challenge.

To efficiently compute OT distances, Cuturi M. [7] introduced a regularized optimal transport distance with entropy constraints,

$$L_M(\mathbf{r}, \mathbf{c}) \stackrel{\text{def}}{=} \min_{\mathbf{P} \in U(\mathbf{r}, \mathbf{c})} \langle \mathbf{P}, \mathbf{M} \rangle + \varepsilon \sum_{i,j} \mathbf{P}_{i,j} (\log(\mathbf{P}_{i,j} - 1)), \quad (3)$$

where  $U(\mathbf{r}, \mathbf{c}) := \{\mathbf{P} \in U(\mathbf{r}, \mathbf{c}) \mid h(\mathbf{P}) \geq h(\mathbf{r}) + h(\mathbf{c}) - \alpha\}$ ,  $h(\cdot)$  is the entropy function, and  $\alpha \in [0, \infty)$ .  $\mathbf{M} \in \mathbb{R}^{D_r \times D_c}$  is the underlying distance matrix, whose element measures the distance between different states.  $\varepsilon$  controls the significance of the entropy regularizer.

In NSTM [50], an encoder parameterized by  $\theta$  is leveraged to generate topic  $z$  from normalized word vector  $\tilde{x}$  by  $z = \text{softmax}(\theta(\tilde{x}))$ . Since  $\tilde{x}$  and  $z$  are two distributions with different support for the same document, in order to learn the encoder, the OT distance is minimized to push  $z$  towards  $\tilde{x}$ , as  $\min_\theta d_M(\tilde{x}, z)$ . Here the cost matrix  $\mathbf{M}$  is specified as the following construction:  $\mathbf{M}_{vk} = 1 - \cos(\mathbf{e}_v, \mathbf{g}_k)$ , where  $\cos$  is the cosine similarity;  $\mathbf{g}_k \in \mathbb{R}^L$  and  $\mathbf{e}_v \in \mathbb{R}^L$  are the embeddings for topic  $k$  and word  $v$ , respectively.  $L$  is the dimension of word embedding and topic embedding. The word embedding  $\mathbf{e}_v$  is obtained by pre-training.  $\mathbf{G} \in \mathbb{R}^{L \times K}$  is used as a set of topic embedding ( $K$  is the number of topics,  $\mathbf{G}$  is random initialized), and also as an optimization parameter, that is, the new optimization objective of OT is,

$$\min_{\theta, \mathbf{G}} L_M(\tilde{x}, z). \quad (4)$$

Combining the OT loss with the traditional cross-entropy loss yields better performance when using either of them. According to the previously mentioned in NTM, encoder  $\theta$  is simulated by a neural network and  $\phi(z)$  is constructed by a single-layer network for decoder. Combining the OT distance with the expected log-likelihood, with Sinkhorn, the optimization objective becomes

$$\max_{\theta, \mathbf{G}} (\varepsilon \tilde{x}^T \log \phi(z) - L_{\mathbf{M}}(\tilde{x}, z)). \quad (5)$$

## 2.2 Manifold Learning

In manifold learning, the observed data is considered a manifestation of one manifold within another, such as low-dimensional data within a manifold in high-dimensional space. Due to the limitations of the internal characteristics of the data, some high-dimensional data can create dimensional redundancy. If the mapping can reduce the data from the high-dimensional space to the low-dimensional space without loss of information, the low-dimensional data may reflect more valuable features of the data.

Therefore, most manifold learning algorithms [2, 5, 6, 8, 10, 12, 14, 18, 19, 32, 35, 39, 40, 42, 44] have a similar general idea, that is: assuming that the data has a certain structural feature in high dimensions, it is expected that the structure can still be maintained after it is reduced to low dimensions.

As a commonly utilized manifold learning algorithm, tSNE [28] measures the distance between sample data as a conditional probability. Suppose there are  $x_i$  and  $x_j$  in the high-dimensional space,  $p_{j|i}$  indicates that the probability that  $x_j$  is in the neighbor of  $x_i$  using Gaussian distribution, the formula is as follows,

$$p_{j|i} = \frac{\exp(-|x_i - x_j|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-|x_i - x_k|^2/2\sigma_i^2)}. \quad (6)$$

The variance  $\sigma_i$  of the Gaussian distribution corresponding to the different points  $x_i$  in high-dimensional space needs to be calculated separately. Similarly, for points in high-dimensional space mapped to the corresponding point in the low-dimensional space (i.e.  $y_i$  and  $y_j$ ), its probability distribution function is defined as

$$q_{j|i} = \frac{\exp(-|y_i - y_j|^2)}{\sum_{k \neq i} \exp(-|y_i - y_k|^2)}. \quad (7)$$

In order to maintain the original relative position information after mapping, we should minimize KL (Kullback-Leibler Divergence) to ensure the similarity between the original high-dimensional distribution ( $P_i$ ) and the mapped low-dimensional distribution ( $Q_i$ ),

$$D = \sum_i \mathbb{KL}(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}. \quad (8)$$

This mapping is not unique; different initial  $y$  can lead to various outcomes. Moreover, this process is not necessarily used for dimension reduction; it can



also be employed for dimension increase or for transferring between manifolds of the same dimension. Other commonly used manifold learning algorithms include Isomap [1], LLE [39], Hessian Eigenmapping [10], etc. Although the methodologies of these algorithms differ, they share similar goals, a detailed discussion of each algorithm is beyond the scope of this text.

### 3 Proposed Model

In this section, we discuss the detailed methodology of DA-NTM. Reiterating the setting mentioned in the background, each document consists of a word count vector  $x \in \mathbb{N}^V$  and is associated with a distribution of  $K$  topics, we denote the distribution as  $z \in \mathbb{R}^K$ , each entry in it represents the proportion of a topic in the document.  $\tilde{x}$  is normalised  $x$  such that  $\tilde{x} := x/\text{length of document}$ . an encoder parameterized by  $\theta$  is utilized to generate the topic  $z$  from normalized word vector  $\tilde{x}$ , represented by  $z = \text{softmax}(\theta(\tilde{x}))$ .

The first challenge we face is that when two representations of a single dataset (word embeddings  $E$  and topic embeddings  $G$ ) are manipulated together, these representations lack a unified metric in their respective metric spaces. This discrepancy poses a difficulty when optimizing NTM with OT. We should set the dimension of the topic embeddings to match that of the word embeddings, and topics learn embeddings suitable for distance comparisons with the word space, while word embeddings are pretrained and fixed. In Fig. 2, we compute the OT distance and manifold learning loss from word embedding to topic embedding. We propose to feed the word embedding with pretrained word embedding GloVe [36]. Topic embedding  $G$  is one optimization parameter so that this cost matrix can be used for the target topic distribution.

Based on the loss function for NSTM  $\max_{\theta, \mathbf{G}} \{\varepsilon \tilde{x}^T \log \phi(z) - L_M(\tilde{x}, z)\}$  in Sect. 2.1, we will impose the constraint of maintaining the distance between the Bag-of-Words representation of different documents and the topic distribution of different documents, and get the loss function like

$$\max_{\theta, \mathbf{G}} (\varepsilon \tilde{x}^T \log \phi(z) - L_M(\tilde{x}, z) + IC(x, z)). \quad (9)$$

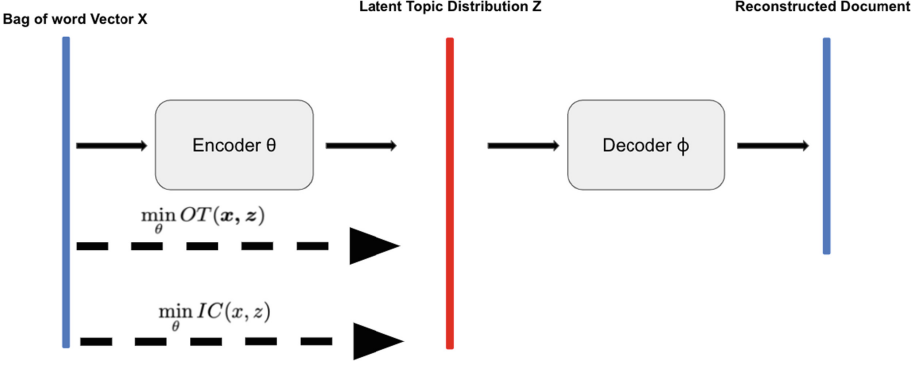
$IC(x, z)$  is the constraint based on the manifold learning algorithm we selected, we term these constraints as isometric constraints because "isometric" in geometry describes transformations or mappings that preserve distances.

If we choose t-SNE as the manifold learning algorithm, and the isometric constrained loss that needs to be added is in the form of

$$IC_{tsne}(x, z) = \sum_i \mathbb{KL}(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{i|j}}{q_{j|i}}, \quad (10)$$

where

$$p_{j|i} = \frac{\exp(-d(x_i, x_j)/2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(x_i, x_k)/2\sigma_i^2)}. \quad (11)$$



**Fig. 2.** Model Structure: Mapping from Bag of Word Vector  $x$  to Latent Topic Distribution  $z$  preserves neighborhood of each document in Bag-of-word vector space

The distance  $d(x_i, x_j)$  between documents can be applied to various metrics (i.e. TF-IDF, WMD (word movement distance) [21] and Euclidean between word counts), but has little impact on topic representation as shown in Table 2.

For different centers  $x_i$ , the variance  $\sigma_i$  of the corresponding Gaussian distribution is also different, and needs to be calculated for each point. Similarly, for the points  $x_i$  and  $x_j$  in the word space mapped to the corresponding points  $z_i$  and  $z_j$  in the topic space, the probability distribution function is as follows,

$$q_{j|i} = \frac{\exp(-\|z_i - z_j\|^2)}{\sum_{k \neq i} \exp(-\|z_i - z_k\|^2)}. \quad (12)$$

It can be observed that Eq. 12 is similar to the computation in Eq. 11, but with the adjustment of setting  $\sigma$  for all points to  $\frac{1}{\sqrt{2}}$  and utilizing the Euclidean distance for the convenience in calculation.

Like NSTM, an encoder is leveraged to generate topic  $z$  from normalised word vector  $\tilde{x}$  by  $z = \text{softmax}(\theta(\tilde{x}))$  and  $\theta$  is a neural network with dropout layer. And the cost matrix  $M$  represents the distance between topic  $k$  and word  $v$ , which is configured as

$$M_{vk} = 1 - \cos(\mathbf{e}_v, \mathbf{g}_k). \quad (13)$$

Now we have the loss function for isometric constrained OT as

$$\max_{\theta, \mathbf{G}} \varepsilon \tilde{x}^T \log \phi(z) - L_M(\tilde{x}, z) + \varepsilon_{ic} \sum_i \mathbb{KL}(P_i || Q_i). \quad (14)$$

$\varepsilon_{ic}$  balanced the coherence and the margins between clusters of topics.

In the previous steps, we exemplified isometric constraints through tSNE. We can employ different manifold learning techniques to design isometric constraints. For instance, if we use LLE to replace tSNE, we should select the  $k$  nearest neighbors of each sample by KNN. Subsequently, we determine the

local reconstruction weight matrix of sample points to find  $w_{ij}$  that minimize  $\sum_i \|x_i - \sum_j w_{ij} x_j\|^2$  and the isometric constraint ( $IC$ ) is  $-\sum_i \|z_i - \sum_j w_{ij} z_j\|^2$ .

Other Distance Awareness NTMs (DA-NTM) corresponding to specific manifold learning algorithms can also be obtained by slightly adjusting the model. The general algorithm is shown in Algorithm 1. Different document distances and various manifold learning algorithms can all be integrated into this algorithm.

## 4 Related Work

NTMs closely related to ours include **ProdLDA** (LDA with Products of Experts) [41], **DVAE** (Dirichlet VAE) [4], **ETM** (Embedding Topic Model) [9] and **WLDA** (Wasserstein LDA) [31]. Instead of using mixture model in LDA, **prodLDA** uses product of experts, then trains the model using AVI. **DVAE** is a neural topic model that obtains the latent topic vector  $z$  by applying a Dirichlet prior/posterior distribution. **ETM**, on the other hand, is a topic model which applied the word embeddings and is obtained through AVI learning. **WLDA** is a topic model based on the Wasserstein AutoEncoders (WAE) framework that minimizes the Wasserstein distance between topic-generated data and real data.

To our knowledge, the works that connect NTMs with manifold learning are still limited. The idea of leveraging visualization-based geometric methods into topic models was first proposed in **SEMAFORE** [24], aiming to combine PLSV [17] and Laplacian Eigenmaps [2]. **DWL** [45] was then proposed for optimal transmission to build topic models, and was further optimized by **OTLDA** [16]. They are all based on the traditional topic models, instead of deep topic models. Compared with their baselines, their accuracy has been improved. However, existing works do not consider preserving geometric properties.

## 5 Experiments

### 5.1 Experimental Settings

**Datasets and Baseline Methods.** We conduct extensive experiments on five benchmark text datasets, including 20 News Groups [22], Web Snippets (WS) [37], Tag My News [38], AG News [48], DBpedia [25].

- AG News is a collection of more than 1 million news articles. AG News has the 4 largest classes (“World”, “Sports”, “Business”, “Sci/Tech”) of AGs Corpus. The AG News contains 30,000 training and 1,900 test samples per class.
- DBpedia is a dataset of information created in the Wikipedia project. It has 14 classes and contains 399,605 training samples and 50,060 test samples.
- The 20 News groups dataset is a collection of 18,846 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. It contains 15,076 training samples and 3,770 test samples.
- Tag My News Dataset is a collection of datasets of short text fragments which are used for topic-based text classifier. It contains 26,077 training samples and 6,520 test samples.

---

**Algorithm 1** Training algorithm for DA-NTM.  $\mathbf{X} \in \mathbb{N}^{V \times B}$  and  $\mathbf{Z} \in \mathbb{R}_{>0}^{K \times B}$  consists of the word count vectors and topic distributions for all the documents, respectively;  $\odot$  is the element-wise multiplication.

**Input:** . Input documents, Pretrained word embeddings  $\mathbf{E}$ , Pretrained document to document distance  $\mathbf{d}(\mathbf{x}_i, \mathbf{x}_j)$ , Topic number  $K$ ,  $\varepsilon$ ,  $\alpha$

**Output:**  $\theta, \mathbf{G}$

- 1: Compute manifold learning parameters in high dimension space (e.g. matrix  $P$  (composed of  $p_{i|j}$  in Eq.11) for tSNE,  $w_{ij}$  for LLE), Randomly initialise  $\theta$  and  $\mathbf{G}$
  - 2: **while** Not converged **do**
  - 3:   Sample a batch of  $B$  input documents  $\mathbf{X}$
  - 4:   Column-wisely normalise  $\mathbf{X}$  to get  $\tilde{\mathbf{X}}$
  - 5:   Compute  $\mathbf{M}$  with  $\mathbf{G}$  and  $\mathbf{E}$  by  $M_{vk} = 1 - \cos(\mathbf{e}_v, \mathbf{g}_k)$
  - 6:   Compute  $\mathbf{Z} = \text{softmax}(\theta(\tilde{\mathbf{X}}))$
  - 7:   Compute the first term of loss  $\varepsilon \tilde{\mathbf{x}}^T \log \phi(z)$
  - 8:    $\Psi_1 = \text{ones}(K, B)/K$ ,  $\Psi_2 = \text{ones}(V, B)/V$  # Sinkhorn iterations #
  - 9:    $\mathbf{H} = e^{-\mathbf{M}/\alpha}$
  - 10:   **while**  $\Psi_1$  changes or any other relevant stopping criterion **do**
  - 11:      $\Psi_2 = \tilde{\mathbf{X}} \odot 1/(\mathbf{H}\Psi_1)$
  - 12:      $\Psi_1 = \mathbf{Z} \odot 1/(\mathbf{H}^T\Psi_2)$
  - 13:   **end while**
  - 14:   Compute the second term of loss  $L_M(\tilde{x}, z)$ :  $L_M = \text{sum}(\Psi_2^T(\mathbf{H} \odot \mathbf{M})\Psi_1)$
  - 15:   Compute manifold learning parameters (e.g. Eq. 11 and Eq. 12 for tSNE);
  - 16:   Compute the third term of loss  $IC(x, z)$  (e.g. IC constraint for tSNE, IC constraint for LLE);
  - 17:   Compute the gradients of loss
  - 18:   Update  $\theta, \mathbf{G}$  with the gradients
  - 19: **end while**
- 

- Web Snippets comprises four abstractive snippet datasets from ClueWeb09, Clueweb12, and DMOZ descriptions. It contains 9,867 training samples and 2,470 test samples.

We compare with the state-of-the-art NTMs, including: ProdLDA [41], Dirichlet VAE (DVAE) [4], Embedding Topic Model (ETM) [9], Wasserstein LDA (WLDA) [31], BERTopic-Doc2Vec [13], NASM [29], Contrastive Learning for Neural Topic Model (CNTM) [34] and NSTM [50].

**Evaluation Metrics.** Since document topic distributions are unsupervised document representations, to assess the quality of such representations, we perform a document clustering task and report the purity and normalized mutual information (NMI) [33], compared to document tags. Using the default train/test split of five datasets, we train the model on the training documents and infer the topic distribution  $z$  on the test documents. In the tSNE visualization mentioned below, since global information about the manifold is required, we will not split the training and testing sets.

Following NSTM [50], we compute purity and NMI (denoted by top-Purity and top-NMI [33]) using the most important topics in test documents as their cluster assignments; We apply the KMeans algorithm to  $z$  of test documents and reports the purity and NMI (denoted by km-Purity and km-NMI) of KMeans clusters, following NSTM [50]. For the first strategy, the number of clusters is equal to the number of topics, while for the second strategy, we vary the number of clusters in KMeans in the range  $\{20, 40, 60, 80, 100\}$ ; We compute AMI (Adjusted Mutual Information) [43] and ARS (Adjusted Rand Score) [15] between most important topics in test documents as their cluster assignments. Adjusted Rand Score (ARS) is used to measure the degree of agreement between the two distributions. The value range is  $[-1, 1]$ . The closer the value is to 1, the better. Mutual Information (Adjusted Mutual Information, AMI) is also used to measure the degree of agreement between the two distributions, the larger the value, the more consistent the clustering effect is with the real situation. We compute topic diversity [9] to measure the variety among the topics generated, ensuring a broad coverage of subjects. We determine the diversity of a topic by calculating the percentage of unique words among the first 25 words of each topic. This approach highlights the distinctiveness of each topic by evaluating the overlap in their most prominent words, thereby offering a measure of the range and diversity of topics produced by the model.

Topic coherence (TC) [23] evaluates the semantic consistency among the most significant words within a topic, utilizing a reference corpus for comparison. To quantify TC, we employ Normalized Pointwise Mutual Information (NPMI), calculated using the Palmetto package. This calculation is based on the first 10 words of each topic, from which we derive an average score across a selection of topics. By adjusting the range of selected topics—from the top 10% with the highest NPMI scores to all topics—we can observe variations in average TC scores, providing insight into the semantic cohesion of topics under different thresholds.

Since document topic distributions are unsupervised document representations, to assess the quality of such representations, we perform a document clustering task and report the purity and normalized mutual information (NMI), compared to document tags.

All the models in comparison ve times with different random seeds and report the mean and standard deviation (as error bars). Unless otherwise specified, we set  $\varepsilon_{ic} = 1$  in experiment.

## 5.2 Quantitative Results

For different models in comparison five times with different random seeds, Table 1 shows the results of top-Purity/NMI with the means and stds. The isometric constrained loss weight  $\varepsilon_{ic} = 1$  and the distance between documents and documents is Euclidean distance between word counts. We can conclude that in the most datasets and metrics, DA-NTM achieves the best performance.

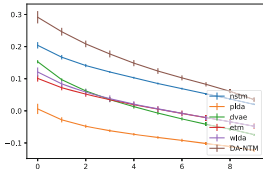
We compared the different results of using different metrics  $d(x_i, x_j)$  to measure the distance between documents and documents on the 20NG dataset in Table 2. Here, we demonstrate that taking different metrics here has little effect

**Table 1.** top-Purity [33] and top-NMI [33] for document clustering. The best scores of each dataset are highlighted in **bold**, with the second-best scores underlined.

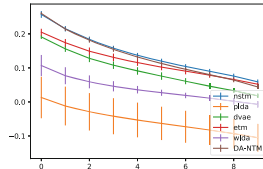
	top-Purity $\uparrow$			top-NMI $\uparrow$		
	20NG	WS	TMN	20NG	WS	TMN
ProdLDA	0.417 $\pm$ 0.004	0.293 $\pm$ 0.023	0.405 $\pm$ 0.157	0.321 $\pm$ 0.004	0.066 $\pm$ 0.016	0.091 $\pm$ 0.101
DVAE	0.281 $\pm$ 0.006	0.284 $\pm$ 0.005	0.477 $\pm$ 0.012	0.187 $\pm$ 0.005	0.059 $\pm$ 0.001	0.113 $\pm$ 0.004
ETM	0.063 $\pm$ 0.003	0.215 $\pm$ 0.001	0.556 $\pm$ 0.022	0.005 $\pm$ 0.005	0.003 $\pm$ 0.003	0.328 $\pm$ 0.010
WLDA	0.117 $\pm$ 0.001	0.239 $\pm$ 0.003	0.260 $\pm$ 0.002	0.060 $\pm$ 0.001	0.026 $\pm$ 0.001	0.009 $\pm$ 0.001
BERTopic	0.319 $\pm$ 0.019	0.328 $\pm$ 0.016	0.491 $\pm$ 0.023	0.368 $\pm$ 0.010	0.151 $\pm$ 0.014	0.202 $\pm$ 0.002
NASM	0.292 $\pm$ 0.009	0.274 $\pm$ 0.037	0.462 $\pm$ 0.013	0.405 $\pm$ 0.025	0.171 $\pm$ 0.014	0.249 $\pm$ 0.077
CNTM	0.334 $\pm$ 0.035	0.411 $\pm$ 0.013	0.523 $\pm$ 0.023	0.401 $\pm$ 0.015	0.131 $\pm$ 0.003	0.302 $\pm$ 0.003
NSTM	<b>0.477</b> $\pm$ 0.011	<u>0.451</u> $\pm$ 0.009	<u>0.637</u> $\pm$ 0.010	<u>0.415</u> $\pm$ 0.012	<u>0.201</u> $\pm$ 0.004	<u>0.334</u> $\pm$ 0.004
DA-NTM	0.334 $\pm$ 0.042	<b>0.635</b> $\pm$ 0.065	<b>0.679</b> $\pm$ 0.028	<b>0.417</b> $\pm$ 0.020	<b>0.461</b> $\pm$ 0.038	<b>0.423</b> $\pm$ 0.018

**Table 2.** Different metrics, including top-Purity [33], top-NMI [33], top-Diversity [9], AMI [43], ARS [15], to measure the different type  $d(x_i, x_j)$  (distance between documents and documents) on the 20NG dataset, with  $\varepsilon_{ic} = 1$

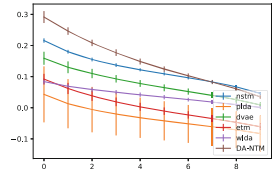
metric	Euclidean between word counts	Euclidean between TF_IDF	word mover distance
top-Purity $\uparrow$	0.334 $\pm$ 0.042	0.283 $\pm$ 0.035	0.347 $\pm$ 0.045
top-NMI $\uparrow$	0.417 $\pm$ 0.020	0.405 $\pm$ 0.016	0.431 $\pm$ 0.027
topic-Diversity $\uparrow$	0.875 $\pm$ 0.007	0.899 $\pm$ 0.012	0.873 $\pm$ 0.003
AMI $\uparrow$	0.410 $\pm$ 0.021	0.386 $\pm$ 0.013	0.391 $\pm$ 0.029
ARS $\uparrow$	0.232 $\pm$ 0.023	0.226 $\pm$ 0.032	0.263 $\pm$ 0.014



(a) 20News



(b) TMN

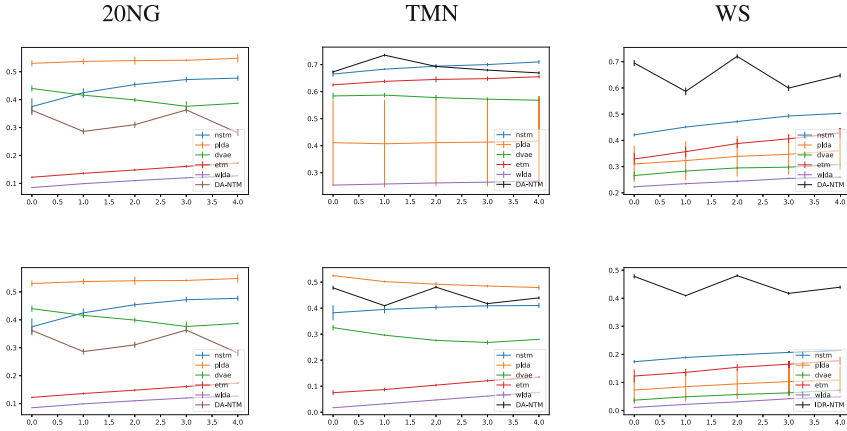


(c) WS

**Fig. 3.** Topic Coherence (TC) [23]: The horizontal axis indicates the proportion of selected topics according to their NPMIs

**Table 3.** Tables compare NSTM and DA-NTM: top-Purity [33], top-NMI [33], top-Diversity [9], AMI [43], ARS [15]

Dataset	20NG		TMN		Webs		AGNews		Dbpedia	
	NSTM	DA-NTM	NSTM	DA-NTM	NSTM	DA-NTM	NSTM	DA-NTM	NSTM	DA-NTM
top-Purity $\uparrow$	0.450 $\pm$ 0.011	0.334 $\pm$ 0.042	0.600 $\pm$ 0.011	0.679 $\pm$ 0.028	0.619 $\pm$ 0.004	0.635 $\pm$ 0.065	0.692 $\pm$ 0.041	0.771 $\pm$ 0.022	0.508 $\pm$ 0.034	0.447 $\pm$ 0.060
top-NMI $\uparrow$	0.403 $\pm$ 0.004	0.417 $\pm$ 0.020	0.348 $\pm$ 0.006	0.423 $\pm$ 0.018	0.364 $\pm$ 0.005	0.461 $\pm$ 0.038	0.323 $\pm$ 0.019	0.431 $\pm$ 0.021	0.457 $\pm$ 0.014	0.556 $\pm$ 0.039
topic-Diversity $\uparrow$	0.556 $\pm$ 0.012	0.875 $\pm$ 0.007	0.921 $\pm$ 0.006	0.923 $\pm$ 0.005	0.921 $\pm$ 0.005	0.899 $\pm$ 0.003	0.884 $\pm$ 0.004	0.916 $\pm$ 0.006	0.857 $\pm$ 0.002	0.878 $\pm$ 0.006
AMI $\uparrow$	0.313 $\pm$ 0.003	0.410 $\pm$ 0.021	0.282 $\pm$ 0.007	0.421 $\pm$ 0.018	0.279 $\pm$ 0.006	0.456 $\pm$ 0.039	0.253 $\pm$ 0.016	0.430 $\pm$ 0.021	0.434 $\pm$ 0.019	0.555 $\pm$ 0.039
ARS $\uparrow$	0.144 $\pm$ 0.008	0.232 $\pm$ 0.023	0.218 $\pm$ 0.018	0.422 $\pm$ 0.010	0.149 $\pm$ 0.009	0.370 $\pm$ 0.063	0.221 $\pm$ 0.047	0.455 $\pm$ 0.028	0.222 $\pm$ 0.022	0.369 $\pm$ 0.046



**Fig. 4.** The first row shows the km-Purity scores [50] and the second row shows the corresponding km-NMI scores [50]. In each subfigure, the horizontal axis indicates the number of KMeans clusters.

on the performances of topic representations. This is due to the fact that the embedding of topics is also our training target. So the inherently uncertain distance function between topics becomes part of our optimized parameters and both the OT loss and the manifold learning loss benefit from it.

In Table 3, we compared DA-NTM and NSTM based on  $\varepsilon_{ic} = 1$  and Euclidean between word counts as document to document distance. It can be observed that DA-NTM outperforms NSTM in most cases.

In Fig. 3, topic coherence is presented with different ranges of selected topics ranging from the top 10% with the highest NPMI scores to all topics. Also, the isometric constrained loss weight  $\varepsilon_{ic} = 1$  and the distance between documents and documents is Euclidean distance between word counts. From Fig. 3, it is evident that DA-NTM significantly outperforms existing methods in terms of topic coherence. This indicates that DA-NTM is indeed adept at capturing the underlying semantic structure of the data.

And Fig. 4 is figure for NMI and purity for different level KMeans clusters. These tables and figures compared the performance of the NSTM and DA-NTM (with Euclidean distance between word counts and  $\varepsilon_{ic} = 1$ ) for different datasets, they verified that adding isometric constraints has improved the performances of the model in metrics for topic quality. This matches our expectation that this model helps for scenarios with insufficient samples.

### 5.3 Visualization Analysis

In order to visually verify that our model has better performance in terms of coherence, we demonstrate it by visualizing the topic representations learned by NSTM and ours with tSNE. For clearer presentation, we sample 1,000 documents of a dataset instead of using all of the documents. We also use colors to indicate

the labels of the documents, to assist understanding. Taking the 20NG dataset as an example in Fig. 5, we can see that in NSTM without the isometric constraint, the topic distributions do not show good clustering characteristics. However, our method with the isometric constrained loss, the learned topic representations appear intuitive clustering structures.

The TMN dataset is a small dataset, then the 2D tSNE dimension reduction of the entire dataset can be plotted more clearly as Fig. 6(a). It can be seen that

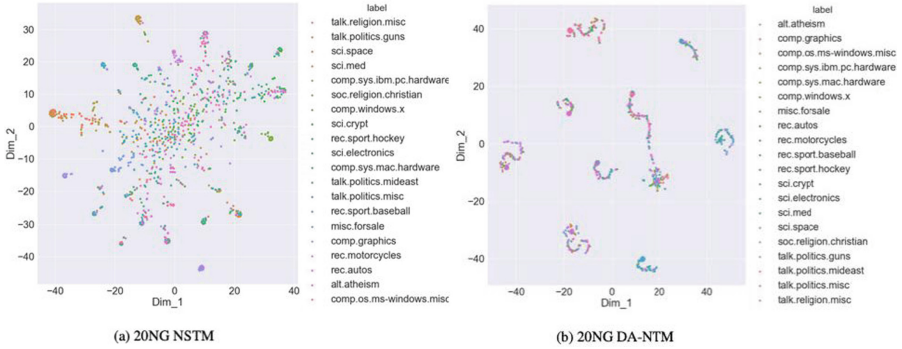


Fig. 5. 20NG Topic representation dimension reduction to 2D by tSNE

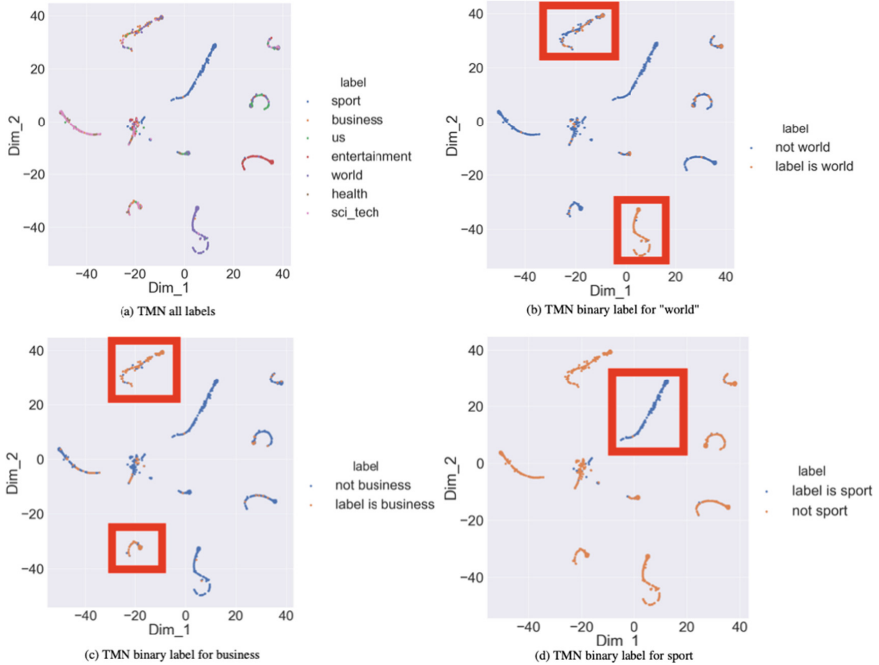


Fig. 6. TMN Topic representation dimension reduction to 2D by tSNE



different labels are obviously assigned to different clusters. In the red boxes of Fig. 6(b), 6(c) and 6(d), we clearly see that a single label can be divided into some specific clusters. The label “business” and label “world” that may share more common vocabularies also share the same cluster in Fig. 6(b) and 6(c). This means that there is indeed a more interpretable connection between document distribution and topic distribution under this framework, and adding isometric constraint can make up for the lack of coherence in past NTMs. More 2D-tSNE visualization results are provided in the Appendix.

## 6 Conclusion

We propose a novel neural topic model that leverages manifold learning and optimal transport in this paper. Specifically, the mechanism of tSNE is integrated to constrain the topic distribution  $z$  and the topic space maintains the geometric properties of the word space. Our model can complement the shortcomings of traditional neural topic models. Extensive experiments have been conducted to demonstrate the unique advantages of the topics derived by our model, which achieves state-of-the-art performance on common metrics of document representation.

In the future, developing visualization tools based on our method can facilitate intuitive exploration of complex datasets. Furthermore, this method can be extended to support multimodal data, such as images and graph data, by integrating these modalities with manifold learning.

## References

1. Balasubramanian, M., Schwartz, E.L.: The isomap algorithm and topological stability. *Science* **295**(5552), 7–7 (2002)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**(6), 1373–1396 (2003)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
4. Burkhardt, S., Kramer, S.: Decoupling sparsity and smoothness in the Dirichlet variational autoencoder topic model. *J. Mach. Learn. Res.* **20**(131), 1–27 (2019)
5. Canas, G., Poggio, T., Rosasco, L.: Learning manifolds with k-means and k-flats. *Advances in neural information processing systems* **25** (2012)
6. Carlsson, G.: Topology and data. *Bull. Am. Math. Soc.* **46**(2), 255–308 (2009)
7. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* **26** (2013)
8. Dasgupta, S., Freund, Y.: Random projection trees and low dimensional manifolds. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, pp. 537–546 (2008)
9. Dieng, A.B., Ruiz, F.J., Blei, D.M.: Topic modeling in embedding spaces. *Trans. Assoc. Comput. Linguistics* **8**, 439–453 (2020)
10. Donoho, D.L., Grimes, C.: Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci.* **100**(10), 5591–5596 (2003)

11. Dumais, S.T.: Latent semantic analysis. *Annual Review of Information Science and Technology (ARIST)* **38**, 189–230 (2004)
12. Genovese, C.R., Perone-Pacifico, M., Verdinelli, I., Wasserman, L.: Manifold estimation and singular deconvolution under hausdorff loss. *Ann. Stat.* **40**(2), 941–963 (2012)
13. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint [arXiv:2203.05794](https://arxiv.org/abs/2203.05794) (2022)
14. Hastie, T., Stuetzle, W.: Principal curves. *J. Am. Stat. Assoc.* **84**(406), 502–516 (1989)
15. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**, 193–218 (1985)
16. Huynh, V., Zhao, H., Phung, D.: Otlida: a geometry-aware optimal transport approach for topic modeling. *Adv. Neural. Inf. Process. Syst.* **33**, 18573–18582 (2020)
17. Iwata, T., Yamada, T., Ueda, N.: Probabilistic latent semantic visualization: topic model for visualizing documents. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 363–371 (2008)
18. Kambhatla, N., Leen, T.: Fast non-linear dimension reduction. *Advances in neural information processing systems* **6** (1993)
19. Kégl, B., Krzyżak, A., Linder, T., Zeger, K.: Learning and design of principal curves. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(3), 281–297 (2000)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
21. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: *International Conference on Machine Learning*, pp. 957–966. PMLR (2015)
22. Lang, K.: Newsweeder: learning to filter netnews. In: *Machine Learning Proceedings 1995*, pp. 331–339. Morgan Kaufmann (1995)
23. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 530–539 (2014)
24. Le, T., Lauw, H.: Manifold learning for jointly modeling topic and visualization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28 (2014)
25. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., Bizer, C.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* **6**(2), 167–195 (2015)
26. Lunga, D., Prasad, S., Crawford, M.M., Ersoy, O.: Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *IEEE Signal Process. Mag.* **31**(1), 55–66 (2013)
27. Ma, L., Crawford, M.M., Tian, J.: Local manifold learning-based  $k$ -nearest-neighbor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **48**(11), 4099–4109 (2010)
28. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of machine learning research* **9**(11) (2008)
29. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: *International Conference on Machine Learning*, pp. 1727–1736. PMLR (2016)
30. Moody, C.E.: Mixing dirichlet topic models and word embeddings to make lda2vec. arXiv preprint [arXiv:1605.02019](https://arxiv.org/abs/1605.02019) (2016)
31. Nan, F., Ding, R., Nallapati, R., Xiang, B.: Topic modeling with wasserstein autoencoders. arXiv preprint [arXiv:1907.12374](https://arxiv.org/abs/1907.12374) (2019)

32. Narayanan, H., Niyogi, P.: On the sample complexity of learning smooth cuts on a manifold. In: COLT (2009)
33. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. *Trans. Assoc. Comput. Linguist.* **3**, 299–313 (2015)
34. Nguyen, T., Luu, A.T.: Contrastive learning for neural topic model. *Adv. Neural. Inf. Process. Syst.* **34**, 11974–11986 (2021)
35. Niyogi, P., Smale, S., Weinberger, S.: Finding the homology of submanifolds with high confidence from random samples. *Discrete Comput. Geometry* **39**(1), 419–441 (2008)
36. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (2014)
37. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: *Proceedings of the 17th International Conference on World Wide Web*, pp. 91–100 (2008)
38. Piwowarski, B., Gallinari, P.: Incremental clustering of news reports. In: *Proceedings of the 29th European Conference on IR Research*, pp. 579–586. Springer (2007)
39. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *science* **290**(5500), 2323–2326 (2000)
40. Smola, A.J., Williamson, R.C., Mika, S., Schölkopf, B.: Regularized principal manifolds. In: *European Conference on Computational Learning Theory*, pp. 214–229. Springer (1999)
41. Srivastava, A., Sutton, C.: Autoencoding variational inference for topic models. arXiv preprint [arXiv:1703.01488](https://arxiv.org/abs/1703.01488) (2017)
42. Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
43. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010)
44. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vision* **70**(1), 77–90 (2006)
45. Xu, H., Wang, W., Liu, W., Carin, L.: Distilled wasserstein learning for word embedding and topic modeling. *Advances in Neural Information Processing Systems* **31** (2018)
46. Yang, X., Zhao, H., Phung, D., Du, L.: Towards generalising neural topical representations. arXiv preprint [arXiv:2307.12564](https://arxiv.org/abs/2307.12564) (2023)
47. Zhang, D.C., Lauw, H.W.: Topic modeling on document networks with Dirichlet optimal transport barycenter. *IEEE Trans. Knowl. Data Eng.* (2023)
48. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: *Advances in Neural Information Processing Systems*, pp. 649–657 (2015)
49. Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., Buntine, W.: Topic modelling meets deep neural networks: a survey
50. Zhao, H., Phung, D., Huynh, V., Le, T., Buntine, W.: Neural topic model via optimal transport. arXiv preprint [arXiv:2008.13537](https://arxiv.org/abs/2008.13537) (2020)



# Ontology-Guided Deep Metric Learning and Applications to Obstetrics

Jules Bonnard<sup>1</sup> , Arnaud Dapogny<sup>2</sup> , Ferdinand Dhombres<sup>3</sup> ,  
and Kévin Bailly<sup>1</sup> 

<sup>1</sup> Sorbonne Université, CNRS, ISIR, 75005 Paris, France  
{jules.bonnard, kevin.bailly}@sorbonne-universite.fr

<sup>2</sup> Apple, Cupertino, USA  
adapogny@apple.com

<sup>3</sup> Sorbonne Université, Université Sorbonne Paris Nord, INSERM, LIMICS, APHP,  
Service de médecine fœtale, GRC26, 75005 Paris, France  
ferdinand.dhombres@inserm.fr

**Abstract.** Obstetrics and gynecology (OB/GYN), branches of medicine that focus on pregnancy and the female reproductive system, heavily rely on ultrasound scanning. The automatic analysis of these images is an interesting tool as it can guide the sonographer in his diagnosis or provide similar images to the sonographer in real time. These tasks have become crucial because of the limited number of experts in the field, but deep learning methods in general have struggled to deal with them because of the lack of large annotated datasets for training. However, leveraging hierarchical rich annotations can be a way to alleviate this problem for learning better structured embedding spaces. In this vein, we propose a Semantic Abstraction Loss (SAL), which guides meta-embeddings to encode the information from the higher-order annotations in a Deep Metric Learning (DML) framework. We then build on the Expert Language Guidance (ELG) introduced by Roth *et al.* [21], that makes use of natural language captions to guide the visual similarities. We therefore propose an Ontology Language Guidance (OLG) that applies this concept to higher-level semantic annotations. Experimentally, we evaluate the impact of the integration of rich annotations through auxiliary embeddings or natural language on two visual similarity datasets: birds classification with CUB-200 and scan plane recognition on SUOG OB/GYN dataset.

**Keywords:** Deep Metric Learning · Medical Imaging · Hierarchical annotations

Ultrasound scanning has become crucial for obstetrics and gynecology tasks (OB/GYN) as it can be used to identify early complications or fetal disorders.

This work was supported by the IUIS institute of Sorbonne University and by the EIT-Health Innovation program (bp2022 #220648) and was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011013628).

The automatic analysis of these images has become critical because of the large number of signs and disorders, as well as the limited number of expert sonographers. In particular, an image retrieval model could allow the ultrasound assistant to offer similarly annotated images to help the sonographer with its diagnosis in real time.

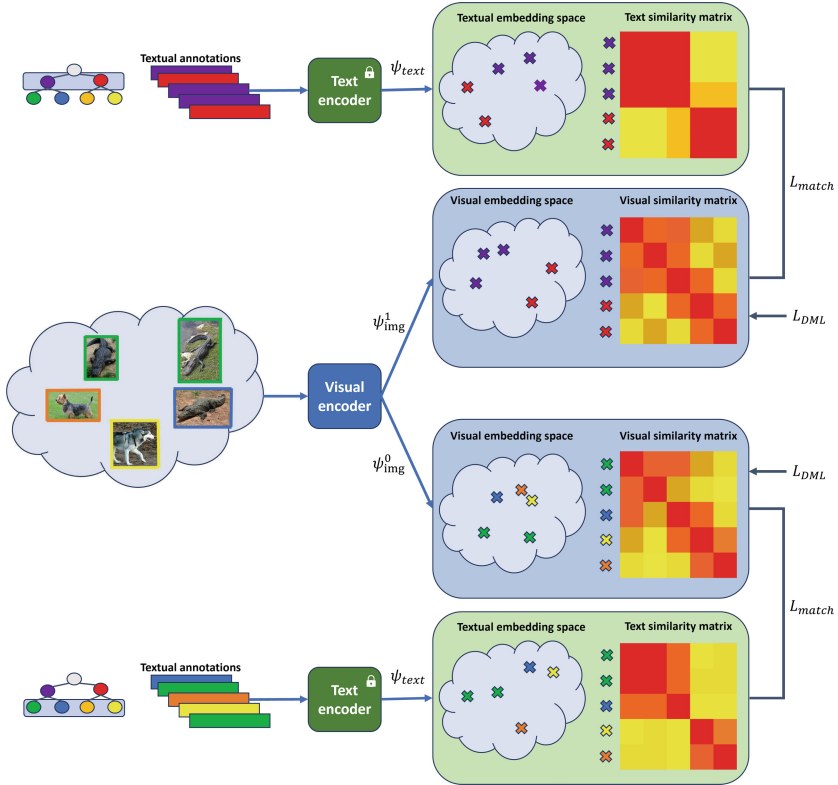
In most of the machine learning problems, the network learns a relatively simple embedding space through a multi-class classification task, as it considers all labels as distinct and exclusive. This is sufficient for tasks trained on extremely large datasets such as ImageNet [10] because of the number of examples and the disparity between classes. However, for problematics such as OB/GYN imaging tasks, where annotated images are scarce, these methods struggle to encode strong semantic distances between training examples since classes are considered equally different (in ImageNet, an *American alligator* is considered equally different to an *African crocodile* than it is to a *Siberian husky*).

We therefore decide to integrate strong semantic information extracted from hierarchical annotations in a deep metric learning (DML) framework. First, we introduce meta-embeddings built to encode semantic information from the higher-order annotations (*animal* and *reptile* as meta-annotations for an *African crocodile*, for instance) and propose a Semantic Abstraction Loss (SAL) designed as a weighted average of DML losses applied at different semantic levels. In order to extend the use of the strong semantic information contained in the rich annotations, we build on the Expert Language Guidance (ELG) introduced by Roth *et al.*, that aims at matching the visual similarities with the textual similarities extracted from the label captions. We propose an Ontology Language Guidance (OLG) built to integrate the textual relations from several abstraction levels in the visual representation model. We show that incorporating robust semantic information, either through meta-labels or textual data, improves the structuring of the latent representation space and contributes to better image similarity performances. Furthermore, this integration aids in producing errors that are semantically closer to the ground truth, which is essential for supporting fetal medicine practices. These experiments are first validated on a classic DML dataset (CUB200 [28]) as well as the SUOG OB/GYN dataset.

Our whole approach is illustrated in Fig. 1. The main contributions in this paper are:

- We propose novel meta-embeddings built to encode the semantic information from different levels of hierarchical annotations. We learn a *Semantic Abstraction Loss (SAL)*, built as a combination of DML losses applied to training pairs of meta-embeddings and meta-classes, in order to improve the latent representation space.
- We propose to integrate rich annotations in the learning framework as textual information to better guide the visual similarity model during training. First, we introduce *rich captions* to make use of the hierarchical structure of the annotations for language guidance. Second, we build on the language guidance module introduced by Roth *et al.* [21] and introduce *Ontology Language*

*Guidance (OLG)* that guides the meta-embeddings’ visual similarity using the textual similarity.



**Fig. 1.** Overview of the proposed method. A visual encoder maps the image to several embedding spaces corresponding to different levels in a semantic hierarchy (2 here for the sake of convenience:  $\psi^0$  and  $\psi^1$ ), where they can be learn through a deep metric learning loss  $L_{DML}$ . We call this term the *Semantic Abstraction Loss* (SAL). Furthermore, textual embeddings can be extracted from the corresponding hierarchical annotation levels through a frozen text encoder, and image-text similarities can be enforced at these different levels through a matching loss  $L_{match}$  in the frame of the proposed Ontology guidance (OLG).

- Experimentally, we evaluate the interest of leveraging higher-order annotations to improve the visual similarity on the CUB-200 [28] dataset and demonstrate the efficiency of the proposed method for assistance in fetal medicine on the SUOG dataset.

## 1 Related Works

First we review existing methods for language guidance in DML. Then, we present methods that leverage structured annotations in this context.

### 1.1 Guiding DML with Natural Language Inputs

Deep Metric Learning aims to learn informative representation spaces that encapsulate rich and significant semantic context, ensuring that embeddings of similar images are close while those of dissimilar images are further away. These frameworks have mostly been used for open-set classification where train and test labels aren't the same (e.g. face verification), and have led to a great interest for tasks such as zero-shot learning ([19, 20, 22, 23]), clustering ([13, 25, 31, 34]) or person re-identification([11, 17, 23]). Multiple methods compare tuples of training samples to improve the semantic context in the visual similarity model. Hadsell *et al.* [14] introduced *Siamese networks*, feature extractors with shared parameters. They optimize a contrastive loss over a pair of training samples that explicitly minimizes the distance between embeddings of similar pairs while maximizing the distance between embeddings of dissimilar pairs. Other works considered triplets [23], quadruplets [8] or n-pairs [24] of samples. However, for these methods to be effective, the training tuples have to be carefully selected in order for the network to successfully learn sound embedding spaces. For instance, tuples that are too easy to separate lead to zeroloss, whereas tuples that are too hard can lead to unstable or collapsed models where all embeddings are pushed towards 0. Therefore, many works focus on finding tuple selection heuristics [7, 23, 26, 30, 31]. Another way to apprehend the DML learning framework is to view it as a classification problem, e.g. using a softmax classifier [5] to separate the classes and use the latent representation during inference. Several approaches have built on this for face recognition [11, 17, 29].

To better represent the semantic relations between training examples, many researchers have explored integrating textual modalities during training. In cross-modal retrieval, the aim is to match the embeddings of the visual and textual inputs in a shared representation space through a discriminative loss [6, 7, 15, 16, 18, 33, 36]. Other methods leverage textual inputs to enhance the visual encoder's predictive capacities. Radford *et al.* [19] introduce CLIP, a method that replaces class annotations by the rich language representations. More specifically, they employ a categorical cross-entropy loss function on the similarity matrix derived from text and image embeddings. In a parallel approach, Roth *et al.* [21] adopt a DML learning framework tailored for relatively modest datasets. They leverage a frozen language encoder to steer the visual similarity matrix towards alignment with the textual similarity matrix.

### 1.2 Leveraging Hierarchical Annotations to Guide the Learning

The aim of integrating hierarchical information is to rectify common classification errors that treat all classes as equally distinct, and therefore enhance the

optimization of inter-class distances. First, several methods map labels to latent representations, with the potential to more effectively encode the semantic similarity between pairs of classes [1, 3, 32]. For instance, Frome *et al.* [12] generate a label representation from a skip-gram language model and then use a ranking loss between the output of a vision model and the label embedding. This allows for the representations of samples from classes that are semantically similar to be closer than with an explicit one-hot labelling. Second, some methods integrate the hierarchical nature of the annotations into the training architecture. Alsallakh *et al.* [2] add classification branches after each convolutional block in the architecture to learn different levels within the class hierarchy. Similarly, Yan *et al.* [35] simplify the class hierarchy by partitioning it into coarse and fine-grained categories. A shared feature extractor is employed to supply inputs to both a coarse component classifier and  $K$  fine component classifiers. However, these methods update their architectures for classification tasks and are specific to certain architectures. This poses a significant challenge, as these specialized architectures require retraining from scratch and do not capitalize on consistent pretraining using a large-scale database. Third, some research has focused on building losses that take into account higher-level semantic context [4, 9, 13, 27]. For instance, Verma *et al.* [27] propose a “context-sensitive loss” where the Lowest Common Ancestor (LCA) extracted from the class hierarchy provides valuable insights to learn similarity metrics between pairs of classes. In contrast to the majority of the studies outlined in this section, our approach is based on DML frameworks rather than classification or regression tasks. This enables robust generalization even to classes not included in the training set, without necessitating a custom architecture or extensive ensemble methods.

## 2 Methodology

In naive DML setups, where all classes are considered exclusive, inter-class distances are not encoded optimally. To solve this, we propose in 2.1 a novel loss  $L_{SAL}$  that integrates hierarchical annotations. In 2.2, we introduce several ways of integrating the rich structured annotations as textual information.

### 2.1 Leveraging Structured Annotations for Image Similarity

Let  $x_i$  denote an image,  $y_i^l$  the class label associated to the image at the  $l$ -th depth of the class hierarchy (with  $y_i^0$  being the leaf, and natural class annotation).

The main idea introduced here is to create meta-embeddings that encode the information issued from the higher order annotations. In the general case, the end-to-end image encoder  $\psi_{img}$  is created as the composition of a common feature extractor  $f$  and a linear projection  $\phi$ . It can be written as:

$$\psi_{img}^0(x_i) = f \circ \phi^0(x_i) \quad (1)$$

We introduce auxiliary *meta-embeddings*  $\psi_{img}^l(x_i)$  output by *meta-projections*  $\phi^l$  built on a single common feature extractor. They can be written as:

$$\psi_{img}^l(x_i) = f \circ \phi^l(x_i) \quad (2)$$



These additional embeddings encapsulate the semantic information conveyed by the meta-annotations. To encourage the feature extraction embedding space to capture these semantic relationships, we introduce a novel loss function  $L_{SAL}$ :

$$L_{SAL} = \sum_{l=0}^L \alpha_l \cdot L_{DML}(\psi^l(x_i), y_i^l) \quad (3)$$

with  $\alpha_l$  the weight associated to each abstraction level  $l$  in the loss and  $L_{DML}$  the DML loss (i.e. triplet loss, margin loss or multi-similarity loss for example). This loss enables the feature extractor  $f$  to acquire features capable of distinguishing among all meta-classes, thus capturing semantic information derived from the extensive class ontology and improving the inter-class distances within the embedding space. This method is illustrated in Fig. 1 (Center-blue part).

To evaluate this method, one only needs the leaf-level *meta embeddings* ( $\psi_{img}^0(x_i)$ ), and therefore does not necessitate any additional information during inference. This method also presents the advantage of being generic and working with different encoder architectures and different DML losses.

## 2.2 Integrating Language Information

We now show how to use the robust textual annotations to guide the visual encoder as illustrated in Fig. 1 (Green part).

**Expert Language Guidance.** An approach to incorporating language guidance within the framework of visual similarity learning, was introduced by Roth *et al.* [21]. In this method, called ELG (short for *Expert Language Guidance*), the authors employ a dual encoder architecture. They harness the rich semantic knowledge acquired by the (frozen) text encoder to influence the image similarity matrix  $S_{img}$  towards mirroring the text similarity matrix  $S_{text}$ . To achieve this, they introduce a novel matching loss:

$$L_{match}(S_{img}, S_{text}) = \frac{1}{B} \sum_i^B \sigma(S_{img}) \log\left(\frac{\sigma(S_{img})}{\sigma(S_{text})}\right) \quad (4)$$

with  $B$  the batch size and  $\sigma$  a row-wise softmax. The final loss term  $L_{ELG}$  is built as a combination of  $L_{match}$  and a classic DML loss that enables the network to learn a structured embedding space. This approach leverages textual information by aligning the visual similarity matrix with the textual similarity matrix. In what follows, we introduce several methods to integrate the semantic hierarchy within DML and image-text matching frameworks.

**Rich Captioning.** The first way to exploit the hierarchical structure of annotations through natural language is to do so through rich captioning. For both works that use language guidance and that are tested in this work, such as CLIP

[19] and Expert Language Guidance [21] (presented in 2.2), we enrich the textual input with hierarchical information.

For instance, there are three levels of annotation hierarchy (species, genus and family) for the CUB-200 dataset [28], a bird classification dataset (presented more extensively in Sect. 3). The caption for the sample  $x_i$  becomes “a photo of a  $y_i^0$  from the genus  $y_i^1$  and the  $y_i^2$  family.”. For instance, for an image of a blue jay, the textual primer used by the model for language guidance would be changed from “a photo of a Blue Jay” to “a photo of a Blue Jay from the genus *Corvidae* and the *Cyanocitta* family”. For the SUOG view dataset, there are only two levels of label hierarchy. Therefore, the additional text for the sample  $x_i$  becomes “ $y_i^0$  from the  $y_i^1$ ”. This approach provides the advantage to adjust the primer according to the specific domain task, offering a degree of freedom to users. However, it also puts all the semantic information at an equal footing.

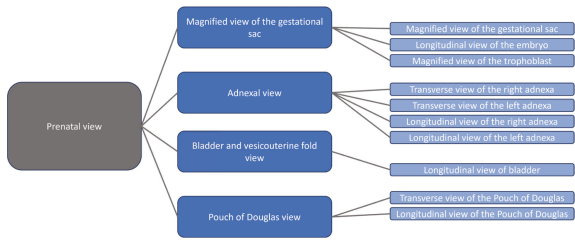
**Language Guidance over Meta Embeddings.** To optimize the enhancement in visual similarity learning facilitated by the *meta embedding learning* presented in Sect. 2.1, we introduce *Ontology Language Guidance* (OLG), where we apply the language guidance loss introduced by Roth *et al.* [21] and presented in 2.2 to the aforementioned *meta embeddings*. We therefore obtain:

$$L_{OLG} = \sum_{l=0}^L \alpha_l \cdot L_{DML}(\psi^l(x_i), y_i^l) + w_l \cdot L_{match}(S_{img}^l, S_{text}^l) \quad (5)$$

In contrast with *rich captioning*, OLG also allows the model to more effectively leverage the hierarchical structure of the annotations by separating the contrinution among all semantic levels. In what follows, we evaluate the interest of guiding a visual similarity framework with higher-level semantic information.



**Fig. 2.** Examples from CUB-200 (top) and SUOG (bottom)



**Fig. 3.** An overview of (a subset of) the view annotations from SUOG ontology.

### 3 Experiments

We present results to experiments led on two datasets: CUB-200 [28] on bird classification and the SUOG OB/GYN dataset for scan plane recognition.

*CUB-200* is a popular DML testbed. It contains 11788 images of birds belonging to 200 different species as a ground-truth class (100 classes in train, 100 classes in test). The first row of Fig. 2 highlights examples from CUB-200. We manually extract the *genus* and *family* to which these species belong from the Avibase world bird database (<https://avibase.bsc-eoc.org/avibase.jsp>) to create higher-level annotations.

*SUOG* The SUOG dataset contains 4323 pregnancy ultrasound images, with 649 used in the test set, randomly sampled to follow the same label distribution as the train set. The second row from Fig. 2 highlights examples from SUOG. We use the *view* annotations as the ground-truth label to perform DML on. There are 18 classes that all belong to a set of 5 *metaclasses*, all extracted from the SUOG ontology created by OB/GYN experts. These classes and metaclasses are shown in Fig. 3.

**Implementation Details.** Unless stated otherwise, we employ a ImageNet-pretrained ResNet50 as the image encoder, and vary the text encoder. The final embedding size of the image encoder is set to 128. We use ADAM with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ , and a batch size of 64. For CUB-200, we use the common data augmentations used in the state-of-the-art methods: the training images are randomly cropped while keeping the same aspect ratio and then are randomly

**Table 1.** Results for the CLIP model on the CUB-200 dataset

Method	Recall@1	std
CLIP	44.98	0.34
CLIP rich caption	45.48	0.37
CLIP + Genus-SAL	46.63	0.2
CLIP + Family-SAL	46.57	0.23
CLIP + Genus-SAL + Family-SAL	<b>47.00</b>	0.4

**Table 2.** Results for the Multism. model on the CUB-200 dataset

Method	Recall@1	Genus-Recall@1	Family-Recall@1
Multism.	63.41 ± 0.45	71.64 ± 0.43	86.16 ± 0.22
Multism. + Genus-SAL	63.36 ± 0.19	<b>71.93 ± 0.26</b>	86.38 ± 0.28
Multism. + Family-SAL	<b>63.50 ± 0.34</b>	71.73 ± 0.50	86.21 ± 0.27
Multism. + Both-SAL	63.22 ± 0.16	71.61 ± 0.30	<b>86.43 ± 0.35</b>

**Table 3.** Results for SAL on the SUOG dataset.

DML method	$\alpha$	Recall@1	Meta-Recall@1
CLIP	0	36.83 $\pm$ 1.58	81.97 $\pm$ 1.56
CLIP + SAL	0.1	37.39 $\pm$ 1.05	<b>82.37 <math>\pm</math> 0.34</b>
CLIP + SAL	0.5	<b>39.17 <math>\pm</math> 1.05</b>	82.16 $\pm$ 1.44
CLIP + SAL	1	38.84 $\pm$ 1.13	81.97 $\pm$ 0.77
Triplet	0	52.32 $\pm$ 1.31	90.91 $\pm$ 0.70
Triplet + SAL	0.1	<b>54.26 <math>\pm</math> 1.14</b>	91.28 $\pm$ 0.58
Triplet + SAL	0.5	<b>54.27 <math>\pm</math> 0.38</b>	91.65 $\pm$ 0.40
Triplet + SAL	1	<b>54.27 <math>\pm</math> 0.69</b>	<b>91.83 <math>\pm</math> 0.50</b>
Softmax	0	55.71 $\pm$ 0.85	91.74 $\pm$ 0.73
Softmax + SAL	0.1	<b>57.29 <math>\pm</math> 0.93</b>	92.17 $\pm$ 0.38
Softmax + SAL	0.5	56.73 $\pm$ 0.65	<b>92.35 <math>\pm</math> 0.45</b>
Softmax + SAL	1	54.73 $\pm$ 0.74	92.29 $\pm$ 0.83
Margin loss	0	53.19 $\pm$ 0.95	89.30 $\pm$ 0.90
Margin loss + SAL	0.1	<b>55.9 <math>\pm</math> 0.88</b>	91.19 $\pm$ 0.65
Margin loss + SAL	0.5	54.3 $\pm$ 0.65	<b>92.02 <math>\pm</math> 0.87</b>
Margin loss + SAL	1	52.36 $\pm$ 0.49	91.96 $\pm$ 0.34
Multisimilarity	0	56.16 $\pm$ 0.81	90.45 $\pm$ 0.83
Multisimilarity + SAL	0.1	<b>56.66 <math>\pm</math> 1.00</b>	91.44 $\pm$ 0.55
Multisimilarity + SAL	0.5	54.85 $\pm$ 0.50	<b>91.96 <math>\pm</math> 0.34</b>
Multisimilarity + SAL	1	54.52 $\pm$ 1.19	91.89 $\pm$ 0.66

flipped. The test images are cropped and centered. For SUOG, only a simple random vertical flip is applied as it allows keeping the ultrasound imaging structure. For the CLIP experiments, a triplet loss term is added to regularize the training, and we use a BERT-small model instead of learning everything from scratch. We also replace the original CCE by a binary cross-entropy loss (BCE).

The main evaluation metric in DML is  $recall@k$  ( $r@k$ ), which equals 1 when at least one of the  $K$  nearest neighbours of a specific query sample shares the same class as that sample, and 0 otherwise. We also present  $meta-recall@k$ , which operates similarly to  $recall@k$  but employs the meta labels as the ground truth.

### 3.1 Guiding the Metric Learning with Prior Meta Annotations

First, we validate the impact of the proposed *Semantic Abstraction Loss* ( $SAL$ ). Table 1 demonstrates that computing visual similarity at meta-class level improves the CLIP model’s predictive performances on the CUB-200 dataset. The addition of  $L_{SAL}$  with both the genus and family information ( $y^1$  and  $y^2$ ) accounts for a 2.02 points global increase, while we can also note that the use of two different levels of hierarchical annotations works better than only

using one, which might indicate that stronger semantic information leads to better embeddings. Table 2 shows a slight performance boost (+0.09 points with family classes) when applying *SAL* loss alongside multisimilarity, which can be explained by the large number of classes and training examples. Another noteworthy aspect of the method is its capability to enable the model to make more insightful errors. Table 2 demonstrates that, on CUB-200, the model trained with *SAL* achieves superior results in terms of genus-r@1 and family-r@1. This implies that even when the nearest neighbours of the query sample do not belong to the same class as the query sample, they may still be semantically close.

Moreover, Table 3 shows that, when applied to 5 different DML losses,  $L_{SAL}$  consistently improves the recall@1 scores (+2.34 points for CLIP, +1.95 points for the triplet loss, +1.58 points for the softmax, +2.69 points for the margin loss and +0.50 points for the multisim loss). It demonstrates that our method is generic to different DML losses. Additionally, integrating *SAL* enhances meta-r@1 by up to 0.99% for CLIP, 0.92% for the triplet loss, 0.61% for the softmax, 1.51% for multisim and 2.72% for the margin loss, echoing the previous results.

### 3.2 Integrating Structured Annotation Through Natural Language

In this section, we show how we leverage rich textual information to improve the performance of DML methods.

**Table 4.** Ablation study of Multisim. and Language Guided method on CUB-200.

Method	ELG	Rich capt.	OLG	Recall@1
Multisim.	✗	✗	✗	63.41 ± 0.45
Multisim.	✓	✗	✗	67.19 ± 0.12
Multisim.	✓	✓	✗	67.33 ± 0.22
Multisim.	✓	✓	Genus+Family ( $\alpha = 0.25$ )	67.5 ± 0.33
Multisim.	✓	✓	Genus+Family ( $\alpha = 0.5$ )	<b>67.74 ± 0.30</b>
Multisim.	✓	✓	Genus+Family ( $\alpha = 0.75$ )	67.61 ± 0.26
Multisim.	✓	✓	Genus+Family ( $\alpha = 1$ )	66.92 ± 0.32
Multisim.	✓	✓	Genus ( $\alpha = 1$ )	66.92 ± 0.25
Multisim.	✓	✓	Family ( $\alpha = 1$ )	67.62 ± 0.35

**Impact of Rich Textual Data During Language-Guided Learning:** We leverage the strong semantic information obtained through the rich annotations using textual representations. We assess the interest of *rich captioning* (see Sect. 2.2) on different datasets. On CUB-200 dataset, Tables 1 and 4 show that using the *rich caption* method slightly improves the predictive performance (0.5 points for CLIP and 0.14 points for multisimilarity). To understand these

**Table 5.** Mean cosine similarity between the embeddings of the rich caption and the simple caption using the CLIP encoder.

Dataset	Level-0 Similarity	Level-1 Similarity	Level-2 Similarity
CUB-200	0.937	0.617	0.655
SUOG	0.937	0.916	—

results, we compare the embeddings of the simple caption and the rich caption. Table 5 show that the mean cosine similarity (over all classes) between the embeddings of the rich caption and the simple caption for the species is very high (0.937), whereas it is much smaller for the genus and family (0.617 and 0.655 respectively). This shows that the text encoding for the rich captions is similar to that of the simple captions, not fully capturing the strong semantic information given by the higher-order annotations, but rather focuses on the most precise terms.

However, we can observe a slight decrease in performance on SUOG in Table 6. This can be explained by the poor performances of the textual encoder on specific OB/GYN terms. Table 5 shows that the rich representations are very close to both the representations of the classes and to the meta-classes. One possibility is that the frozen text encoder embeds all the SUOG classes and meta-classes in a very tight region because of its lack specific OB/GYN knowledge.

**Guiding Meta Embeddings Using Natural Language:** As we have previously illustrated the utility of integrating the  $SAL$  loss term within a DML framework, we hereby prove the efficacy of leveraging natural language to guide these meta-embeddings. For the CUB-200 dataset, results in Table 4 show that adding the OLG loss term helps improve the model’s predictive performance, as it helps structure the embedding space and therefore improve the inter-class distances. When the model is guided using both genus and family annotations along with language cues, the performance improves from 63.41% to 67.74% at its peak, with the OLG loss term contributing to a 0.55-point enhancement compared to the model employing basic ELG language guidance alone. Table 6 shows that employing basic ELG marginally enhances the  $r@1$  by only 0.17 points when directed by a straightforward caption. As anticipated, the inclusion of an  $L_{SAL}$  loss term elevates the  $r@1$  to 56.49, accompanied by a 0.43-point augmentation in meta- $r@1$ , confirming earlier findings. Additionally, guiding the model with OLG demonstrates a slight enhancement in results, with  $r@1$  reaching 56.61%. Nevertheless, we can see that the results achieved with OLG still fall slightly short of those obtained solely with  $L_{SAL}$ . These findings highlight that language guidance is beneficial only when the language model offers good context to distinguish classes. In the case of SUOG, the text encoder even marginally degrades the model’s performance because of its lack of specialized knowledge.

Hence, using textual data to guide a visual DML model only yields interesting results provided that the text encoder comprehensively understands the semantic information encapsulated within the training data.

**Table 6.** Ablation study on SUOG dataset for language guided meta-learning.

DML method	$SAL \alpha$	Rich capt.	ELG	OLG	r@1	meta-r@1
Multisim.	✗	—	✗	✗	$56.16 \pm 0.81$	$90.45 \pm 0.83$
Multisim.	✗	✗	✓	✗	$56.33 \pm 0.98$	$90.17 \pm 0.54$
Multisim.	✗	✓	✓	✗	$56.21 \pm 1.31$	$90.35 \pm 0.89$
Multisim.	✓	✗	✓	✗	$56.49 \pm 0.14$	<b><math>90.88 \pm 0.86</math></b>
Multisim.	✓	✗	✓	✓	<b><math>56.61 \pm 0.77</math></b>	$90.57 \pm 0.79$
Multisim.	✓	—	✗	✗	<b><math>56.66 \pm 1.00</math></b>	<b><math>91.96 \pm 0.34</math></b>

## 4 Conclusion

In this paper, we investigated the incorporation of rich annotations to enhance a deep metric learning framework. We first leveraged the hierarchical annotations extracted from the class ontology by creating auxiliary *meta-embeddings* that are pushed to encode different levels of meta-annotations with the novel  $L_{SAL}$  loss.  $L_{SAL}$  enables the model to better encode inter-class relations, bringing closer samples from classes that share the same metaclass. Second, we capitalized on the robust textual information associated with the annotations and proposed *Ontology Language Guidance (OLG)*, a method that specifically guides the *meta-embeddings* using natural language. A notable advantage of these methods is that although they require supplementary input information in the form of meta-annotations during training, they do not require it during inference.

We validated the interest of SAL and OLG on the CUB-200 dataset and SUOG OB/GYN dataset. In particular, we showed that the rich captioning provided a limited improvement on both datasets, as the text encoders mainly focused on the leaf-level classes in the rich captions. The integration of *OLG* addressed this concern by optimizing the influence of all levels of annotations. While it demonstrated effectiveness on CUB-200, where the text encoder effectively captured semantic context, it did not enhance the model’s representational capacity on the SUOG dataset. This may be due to the text encoders’ inability to distinguish between classes and metaclasses due to a domain gap.

As a conclusion, it is interesting to note that guiding a DML model using rich annotations, whether it be through auxiliary embeddings with  $L_{SAL}$  or language guidance with rich captioning or OLG, attest to generally improve the representations given by the model. However, it is usually more interesting to use textual representations when the text encoder can provide semantic context in the input domain, which is a future direction that we would like to investigate.

**Acknowledgements.** This work has been supported by ANR (FacIL, project ANR-17-CE33-0002), by the IUIS institute of Sorbonne Univ. and by the EIT-Health Innovation program (bp2022 #220648)

## References

1. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2927–2936 (2015). <https://doi.org/10.1109/CVPR.2015.7298911>
2. Alsallakh, B., Jourabloo, A., Ye, M., Liu, X., Ren, L.: Do convolutional neural networks learn class hierarchy? *IEEE Trans. Visual. Comput. Graph.* **24**, 152–162 (2017). <https://api.semanticscholar.org/CorpusID:192425>
3. Barz, B., Denzler, J.: Hierarchy-based image embeddings for semantic image retrieval. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 638–647 (2019). <https://doi.org/10.1109/WACV.2019.00073>
4. Bertinetto, L., Mueller, R., Tertikas, K., Samangooei, S., Lord, N.A.: Making better mistakes: leveraging class hierarchies with deep networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12503–12512. IEEE Computer Society, Los Alamitos, CA, USA (2020). <https://doi.org/10.1109/CVPR42600.2020.01252>, <https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01252>
5. Bridle, J.: Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: Touretzky, D. (ed.) *Advances in Neural Information Processing Systems*, vol. 2. Morgan-Kaufmann (1989)
6. Cao, W., Lin, Q., He, Z., He, Z.: Hybrid representation learning for cross-modal retrieval. *Neurocomputing* **345**, 45–57 (2019). <https://doi.org/10.1016/j.neucom.2018.10.082>, <https://www.sciencedirect.com/science/article/pii/S0925231219301407>, deep Learning for Intelligent Sensing, Decision-Making and Control
7. Carvalho, M., Cadène, R., Picard, D., Soulier, L., Thome, N., Cord, M.: Cross-modal retrieval in the cooking context: learning semantic text-image embeddings. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018). <https://api.semanticscholar.org/CorpusID:13755946>
8. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1320–1329 (2017). <https://api.semanticscholar.org/CorpusID:14795862>
9. Deng, J., Berg, A.C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *Computer Vision - ECCV 2010*, pp. 71–84. Springer, Berlin Heidelberg, Berlin, Heidelberg (2010). <https://doi.org/10.1007/978-3-642-15555-0-6>
10. 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)
11. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4685–4694 (2019). <https://doi.org/10.1109/CVPR.2019.00482>



12. Frome, A., et al.: DeViSE: a deep visual-semantic embedding model. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 26. Curran Associates, Inc. (2013)
13. Ge, W., Huang, W., Dong, D., Scott, M.R.: Deep metric learning with hierarchical triplet loss. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision - ECCV 2018*, pp. 272–288. Springer International Publishing, Cham (2018). [https://doi.org/10.1007/978-3-030-01231-1\\_17](https://doi.org/10.1007/978-3-030-01231-1_17)
14. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006) **2**, 1735–1742 (2006). <https://api.semanticscholar.org/CorpusID:8281592>
15. He, Y., Xiang, S., Kang, C., Wang, J., Pan, C.: Cross-modal retrieval via deep and bidirectional representation learning. *IEEE Trans. Multimedia* **18**, 1363–1377 (2016). <https://api.semanticscholar.org/CorpusID:22518199>
16. Huang, X., Peng, Y.: Cross-modal deep metric learning with multi-task regularization. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 943–948 (2017). <https://api.semanticscholar.org/CorpusID:7884673>
17. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: SphereFace: deep hypersphere embedding for face recognition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6738–6746 (2017). <https://api.semanticscholar.org/CorpusID:206596594>
18. Peng, Y., Qi, J., Yuan, Y.: CM-GANs: cross-modal generative adversarial networks for common representation learning. *abs/1710.05106* (2017). <https://api.semanticscholar.org/CorpusID:8355505>
19. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning* (2021). <https://api.semanticscholar.org/CorpusID:231591445>
20. Roth, K., Brattoli, B., Ommer, B.: MIC: mining interclass characteristics for improved metric learning. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7999–8008 (2019). <https://api.semanticscholar.org/CorpusID:202749912>
21. Roth, K., Vinyals, O., Akata, Z.: Integrating language guidance into vision-based deep metric learning. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16156–16168 (2022). <https://doi.org/10.1109/CVPR52688.2022.01570>
22. Sanakoyeu, A., Tschernezki, V., Buchler, U., Ommer, B.: Divide and conquer the embedding space for metric learning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 471–480. IEEE Computer Society, Los Alamitos, CA, USA (2019). <https://doi.org/10.1109/CVPR.2019.00056>, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00056>
23. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015)
24. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc. (2016)
25. Sohn, K., Shang, W., Yu, X., Chandraker, M.: Unsupervised domain adaptation for distance metric learning. In: *International Conference on Learning Representations* (2018). <https://api.semanticscholar.org/CorpusID:108299626>

26. Suh, Y., Han, B., Kim, W., Lee, K.M.: Stochastic class-based hard example mining for deep metric learning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7244–7252 (2019). <https://doi.org/10.1109/CVPR.2019.00742>
27. Verma, N., Mahajan, D., Sellamanickam, S., Nair, V.: Learning hierarchical similarity metrics. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2280–2287 (2012). <https://doi.org/10.1109/CVPR.2012.6247938>
28. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
29. Wang, H., et al.: CosFace: large margin cosine loss for deep face recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 5265–5274 (2018). <https://api.semanticscholar.org/CorpusID:68589>
30. Wang, X., Han, X., Huang, W., Dong, D., Scott, M.R.: Multi-similarity loss with general pair weighting for deep metric learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5017–5025 (2019). <https://api.semanticscholar.org/CorpusID:118646482>
31. Wu, C.Y., Manmatha, R., Smola, A.J., KrÄhenbÄhl, P.: Sampling matters in deep embedding learning. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2859–2867 (2017). <https://doi.org/10.1109/ICCV.2017.309>
32. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 69–77 (2016)<https://doi.org/10.1109/CVPR.2016.15>
33. Xu, X., He, L., Lu, H., Gao, L., Ji, Y.: Deep adversarial metric learning for cross-modal retrieval. *World Wide Web* **22**, 657–672 (2019). <https://api.semanticscholar.org/CorpusID:4560834>
34. Yan, J., Luo, L., Deng, C., Huang, H.: Unsupervised hyperbolic metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12465–12474 (2021)
35. Yan, Z., et al.: HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In: 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2740–2748 (2014). <https://api.semanticscholar.org/CorpusID:206770495>
36. Zhen, L., Hu, P., Wang, X., Peng, D.: Deep supervised cross-modal retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)



# CoFE: Consistency-Driven Feature Elimination for eXplainable AI

Revoti Prasad Bora<sup>1</sup>(✉), Philipp Terhörst<sup>2</sup>, Raymond Veldhuis<sup>1</sup>,  
Raghavendra Ramachandra<sup>1</sup>, and Kiran Raja<sup>1</sup>

<sup>1</sup> Norwegian University of Science and Technology, Gjøvik, Norway  
{revoti.p.bora,raymond.veldhuis,raghavendra.ramachandra,  
kiran.raja}@ntnu.no

<sup>2</sup> University of Paderborn, Paderborn, Germany  
philipp.terhoerst@uni-paderborn.de

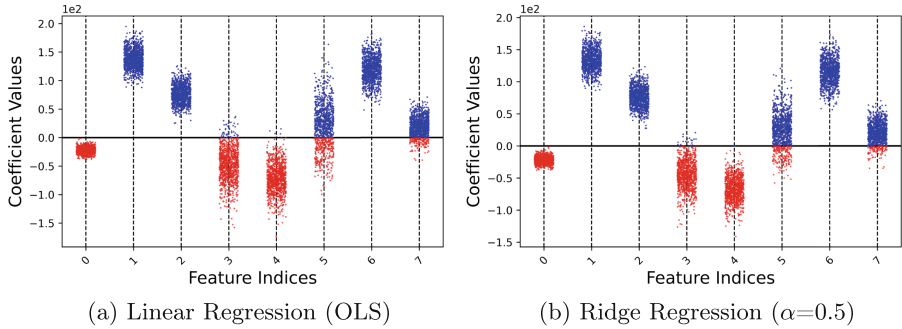
**Abstract.** Feature selection is crucial in building Machine Learning (ML) models. A model trained on selected features can outperform models trained on all available features in predictive accuracy. Most feature selection algorithms focus on the predictive accuracy of the models. Hence, feature selection algorithms incorporate various statistical methodologies to maximize predictive accuracy. While this serves the purpose of maximizing accuracy, consistency is needed to explain the model's decision. A regression model can be useful for eXplainable Artificial Intelligence (XAI) if and only if the coefficients show consistent signs (positive or negative) despite the inherent variability in data. This work demonstrates that linear regression models built using features selected by traditional approaches exhibit poor consistency of coefficient signs. This inconsistency in the sign of coefficients can hinder the understanding of feature influence on the target. To address this, we propose a novel feature selection algorithm that selects only those features that minimize the fluctuation of the model's coefficients' sign, i.e., consistent features. Our experimental results on three different public datasets and two regression techniques demonstrate the effectiveness of our approach. Thus, models built on the selected features using our approach exhibit better consistency of the coefficients' sign than models built on features selected using traditional, with minimal impact on predictive accuracy. This substantial improvement in consistency shows that our approach cannot only compete with existing approaches in terms of accuracy but also outperform them in terms of consistency, making it a valuable tool for XAI applications.

## 1 Introduction

Machine Learning (ML) models are widely used across multiple domains to solve various problems [1, 3, 17, 28]. In highly regulated fields like health care,

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_24](https://doi.org/10.1007/978-3-031-78189-6_24).



**Fig. 1.** Variation of coefficients' sign of models (OLS and Ridge Regression) trained on 1000 bootstrapped datasets of Diabetes Dataset. Red color indicates coefficients with negative values (i.e.  $< 0$ ) while the blue color indicates the coefficients with positive values (i.e.  $\geq 0$ ).

insurance, law enforcement, etc. there has been concerns regarding fairness, privacy and trustworthiness of ML models [1, 28]. Some researchers have therefore advocated the use of interpretable models as compared to black-box models for high stakes decision making scenarios [28]. Further, techniques like LIME [27] and its variants (D-LIME [32], ALIME [30], S-LIME [34], BayLIME [33]), use an interpretable model to explain complex decision boundary in local scope. Thus interpretable models (like Linear Regression, Decision Trees etc.) are used extensively in situations where the explanation for the predicted outcome is desired.

Among the interpretable models, Linear Regression with Ordinary Least Squares (OLS) and Ridge Regression are common choices and the focus of our work. The coefficients of Linear models<sup>1</sup>, which are used as an explanation, should exhibit consistency<sup>2</sup> in the sign of the coefficients for similar inputs [2]. Thus, from an XAI standpoint, the consistency of the coefficients' sign for small variations in the input is crucial. Although sensitivity analysis of Linear Regression was studied in several works [4, 7, 11, 15], yet the consistency (i.e., robustness to sign flips) of the coefficients has not been addressed in the literature from an XAI perspective, thus overlooking a critical aspect of explainability. We exemplify the inconsistency of coefficients of both OLS and Ridge regression models on Diabetes Dataset [9] in Fig. 1a and Fig. 1b. The inconsistency was noted across the models trained by bootstrapping the Diabetes Dataset [9] 1000 times for both the OLS and Ridge models. Specifically, the coefficients corresponding to feature indices 3, 5, and 7 in Fig. 1a and feature indices 5 and 7 in Fig. 1b show high variability in their signs, i.e., these features contribute to the model's out-

<sup>1</sup> We will refer to Linear Regression with OLS and Ridge Regression collectively as linear models unless otherwise stated further-on in the paper.

<sup>2</sup> From hereon, we will use the term consistency to denote the stability of the sign of the coefficients. Along the same lines, a consistent feature would mean that the corresponding coefficient, in the linear model, is robust to sign flips despite the inherent variability of data.

put (i.e., target) positively in some cases and negatively in others, making them inconsistent. These coefficients violate the consistency property (i.e., flipping of coefficients' sign) as mentioned by [2].

In this paper, we hypothesize that features which lead to inconsistent signs of coefficients can be removed before training the model, i.e., in the feature selection stage. We, therefore, propose a novel feature selection (elimination) approach to remove these inconsistent features. We then compare the consistency and the predictive performance of the models trained on the selected features by the proposed approach vs. those trained using features from traditional approaches. Our experiments provide robust statistical evidence that the proposed feature selection approach demonstrates substantial gain in consistency with considerably low loss in predictive accuracy. In Sect. 2, we discuss the traditional feature selection approaches, their limitations, and our contribution; in Sect. 3, we introduce our novel feature selection approach through elimination. The results are presented in Sect. 5 followed by conclusions in Sect. 7.

## 2 Background

Feature selection approaches can be divided into three main types viz. Filter, Wrapper and Embedded [12]. Filter approaches work in selecting relevant features based on the predictors without involving the target. Wrapper approaches use machine learning model as a black box to estimate predictive power of features and then select a subset. While embedded approaches are tightly coupled with the concerned ML algorithm and perform feature selection as a part of model training [12]. Recursive Feature Elimination (RFE) and three flavors of Sequential Feature Selections, viz. Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), and Bidirectional feature (BD) selection have been popularly used in machine learning [6, 16, 19].

RFE is a wrapper-based feature selection approach [12] that aims to recursively remove the least important feature(s) from the set, based on the weights assigned by a particular machine learning model, until the desired number of features is achieved. A model is trained on the initial set of features and the importance of each feature is obtained either directly from the model (like coefficients in linear models) or through a predefined metric (e.g., accuracy). The least important feature(s) is/are pruned set iteratively until the desired number of features are retained. SFS is a wrapper-based sequential feature selection approach that incrementally builds a model by adding features one at a time until a stopping criterion is met [10, 20, 26]. Forward selection typically follows iterations over remaining features (starting from empty set) and build feature set to improve the model's performance the most until a predefined number of features or no further improvement in model performance is seen within the number of preset iterations. SBS also known as backward elimination, is another wrapper-based sequential feature selection approach that starts with a full set of features and removes one feature at a time until a stopping criterion is met [10, 20]. Backward selection typically follows iteration over the features (starting

from full set) and remove the one that contributes the least to the model’s performance until a stopping criterion is reached (e.g., a predefined number of features or no further improvement in model performance). BD, also known as stepwise selection, combines forward and backward selection approaches. It starts with an empty set of features and alternates between forward and backward steps until a stopping criterion is met [10,25].

## 2.1 Limitations of Current Approaches

RFE, SFS, SBS, and BD are popular feature selection approaches, each with a unique approach to ranking and selecting the most relevant features for predictive modeling. However, in the specific context of linear models, these approaches prioritize predictive accuracy without accounting for signs’ stability (consistency). As such, the coefficients’ sign could have a high variance for small changes in the training data as shown in Fig. 1. Any model, for instance, a Linear Regression model based on the selected features, is consistent if and only if the contribution of features, i.e., coefficients in the trained model, remains consistent despite the inherent variability of data. Thus, a coefficient exhibiting frequent sign flips leads to ambiguity in determining the direction of impact (positive or negative) on the target. However, none of the existing works on feature selection focused on selecting consistent (low variance in the sign of coefficients) features. Hence, despite having good predictive power, these approaches are unusable for XAI applications where the feature’s impact has to be explained for a decision [2].

## 2.2 Our Contributions

We consider the problem of consistent feature selection (i.e., inconsistent feature elimination) and hypothesize that inconsistent features (i.e. prone to coefficient sign flips) can be removed before training and thus, making the model coefficients robust to sign flips. This would significantly enhance the explainability with marginal degradation in the predictive accuracy of the model. We consider wrapper based family in this work to validate our idea as we intend to provide a feature selection approach that can be used in conjunction with widely employed linear models. Our approach relies on estimating the inconsistency of features (uncertainty associated with the sign of the features) and eliminating the inconsistent features to make the model explainable. We consider a case study of regression (with linear and ridge regression) in linear models for demonstrating our proposed approach of selecting consistent features. We demonstrate the inability of traditional wrapper based feature selection approaches to select consistent features on three well known public datasets (i.e., House Prices - Advanced Regression Techniques dataset from Kaggle [18], Superconductivity [13] and Appliance Energy Prediction [5] datasets from UCI repository). We further show that the proposed approach can compete against existing feature

selection approaches by providing similar predictive performance while significantly outperforming in terms of explainability metrics (i.e. selecting consistent features).

### 3 Proposed Feature Selection Approach

We propose a novel feature selection algorithm under the family of wrapper-based feature selection. Our approach estimates the sign entropy of coefficients (refer Equation (1)) and uses it to eliminate inconsistent features. When excluded from the dataset, these eliminated features make the models consistent for explanations. As our approach is a backward feature selection approach (i.e., feature elimination) and is designed to offer consistency of explanations, we refer to it as Consistency-driven Feature Elimination (CoFE).

In lines with the family of wrapper based approaches, CoFE is an iterative approach and it begins the first iteration by bootstrapping the dataset  $D$  and building models on the bootstrapped datasets. The coefficients from these trained bootstrapped models are used to estimate the sign entropy of the coefficients using Kernel Density Estimate (KDE). We use Scott's rule of thumb [29] to calculate the bandwidth in all KDE calculations. Thus, features with positive sign entropy are considered inconsistent, while features with zero sign entropy are deemed to be consistent. We use the term sign entropy to quantify the consistency of a single feature and the term Coefficient Sign Stability (CoSS) to measure the consistency of all features in a model (details in Sect. 3.1).

In the subsequent iterations, the features with positive sign entropy are removed (i.e., dropped) to eliminate their contribution in predicting the target (i.e.,  $Y$ ), and the sign entropy of the remaining features is re-estimated. This process is continued iteratively until a subset of consistent features is obtained (or alternatively for a set of a predefined number of iterations). If none of the features were identified as consistent, then the algorithm can be initiated with a higher threshold of sign entropy instead of zero to select relatively stable features than the original set of features (see Algorithm 1 for details).

#### 3.1 Coefficient Sign Stability (CoSS)

We propose a metric, Coefficient Sign Stability, to quantify the variability in the sign of the coefficients of a model. It reflects the degree of inconsistency in the direction of the effect of the explanatory variables for the inherent variability of data. A lower value of CoSS would indicate that the coefficients of a model are more consistent. In contrast, a higher value of CoSS would indicate that the sign of the coefficients of a model is likely to flip from positive to negative for small variations in the data. CoSS for model 'M' and feature selection technique 'k' can be shown as below:

$$CoSS_M^k = \frac{1}{N} \sum_i H(\text{sign}_i) \quad (1)$$

**Algorithm 1.** Consistency-driven Feature Elimination (CoFE)

---

```

1: Input: Data  $D$ , max_tolerance  $\tau$ , number of bootstrap samples  $N = 1000$  max
   iterations  $iter\_max = 10$ 
2: Output: Indices of selected features  $sel\_indices$ 
3: Initialize: Tolerance counter  $t = 0$ ,  $sel\_indices = col\_ids(D)$ 
4: while  $t < \tau$  OR  $iter < iter\_max$  do
5:   Initialize a matrix  $\mathbf{C}'$  with dimensions  $N \times D$ 
6:   ▷  $D$ : number of features in dataset
7:   for  $i = 1$  to  $N$  do
8:     Bootstrap sample  $D_i$  from  $D$ 
9:     Fit regression model on  $D_i$  to get coefficients  $\mathbf{c}_i$ 
10:     $\mathbf{C}'[i, :] \leftarrow \mathbf{c}_i$ 
11:    Compute the sign entropy  $H$  for each coefficient across  $N$  models using KDE
12:     $F' \leftarrow$  coefficients with  $H > 0$ 
13:    if  $|F'| \neq 0$  then
14:       $D \leftarrow D[:, \neg F']$ 
15:       $sel\_indices \leftarrow sel\_indices[:, \neg F']$ 
16:       $t \leftarrow 0$ 
17:    else
18:       $t \leftarrow t + 1$ 
19:     $iter \leftarrow iter + 1$ 
20: return  $\mathbf{C}$ 

```

---

where,

$$H(\text{sign}_i) = -p_i^+ \log_2(p_i^+) - p_i^- \log_2(p_i^-)$$

is the sign entropy of the  $i^{\text{th}}$  coefficient and the quantities  $p_i^+$  and  $p_i^-$  are the probabilities of the  $i^{\text{th}}$  coefficient to be positive or negative respectively.  $p^+$  and  $p^-$  are calculated for each coefficient by using Kernel Density Estimation (KDE) owing to its non-parametric nature [29].

## 4 Experimental Setup

We conduct a series of experiments on three different publicly available datasets that include House Prices - Advanced Regression Techniques dataset from Kaggle [18], Superconductivity [13] and Appliance Energy Prediction [5] datasets from UCI repository. These would be referred to as Housing, Superconductivity and Energy datasets respectively. Each of these datasets are used in the context of regressing the price, conductivity value and energy consumption respectively. We use OLS [24] and Ridge Regression [14]<sup>3</sup> with our proposed approach to show the impact of consistent feature selection<sup>4</sup>.

<sup>3</sup> Our code makes use of implementations in the official Pypi repositories (<https://pypi.org/project/feature-selector/> and <https://pypi.org/project/mlxtend/>) for all feature selection algorithms to support reproducibility.

<sup>4</sup> We do not consider LASSO regression in this work for comparison as it uses feature selection a part of the training process making it a joint/embedded feature selection and regressor algorithm [12, 31].



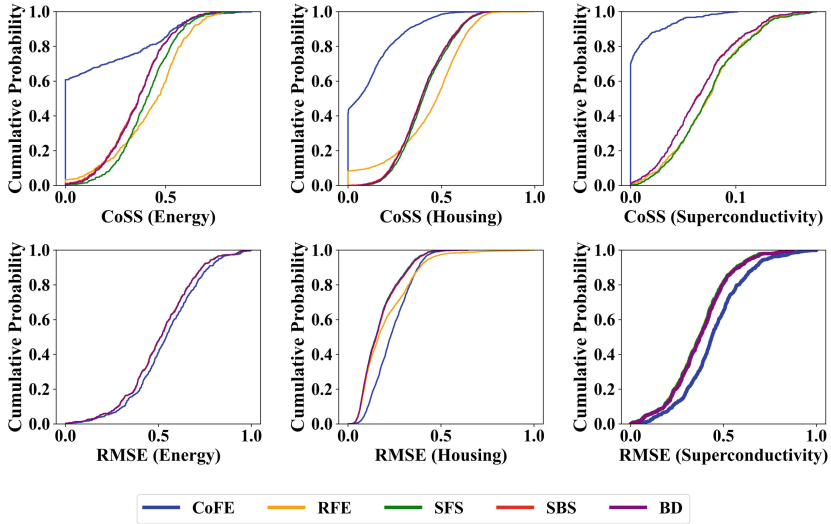
Further, to eliminate the influence of any highly correlated features in the experiments, we discard all highly correlated (i.e. with a Pearson Correlation  $\geq 0.8$ ) features from the datasets. We report our results from k-fold ( $k = 5$ ) cross-validation with 30 repeats to calculate the CoSS and RMSE (details in Section Section S1.2) values. The RMSE values are normalized to a range of  $[0,1]$  using min-max scaling Section S1.1. We compare our results against RFE, SFS, SBS and BD which are feature selection approaches based on ranking. Hence, for a fair comparison with our proposed CoFE as a feature selection approach Algorithm 1) we determine the optimal number of features/subset (i.e. ‘n’ top features) required for maximum predictive accuracy for each of the counterpart approaches using a k-fold cross-validation with  $k = 5$ .

**Table 1.** Median CoSS scores and RMSE of all feature selection approaches for OLS and Ridge regression. Lower CoSS and RMSE scores is good.

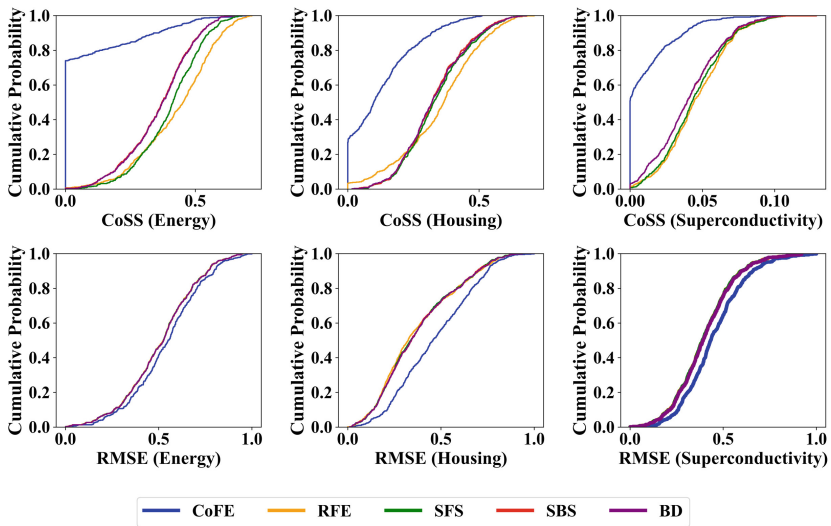
M	CoSS					RMSE				
	CoFE	RFE	SFS	SBS	BD	CoFE	RFE	SFS	SBS	BD
Housing Price Dataset										
OLS	<b>0.06</b>	0.48	0.4	0.39	0.39	0.23	0.16	0.15	0.15	0.15
R0.1	<b>0.09</b>	0.45	0.38	0.37	0.37	0.32	0.23	0.23	0.23	0.23
R0.5	<b>0.1</b>	0.37	0.33	0.32	0.32	0.46	0.32	0.34	0.34	0.34
R0.9	<b>0.1</b>	0.33	0.31	0.31	0.31	0.48	0.36	0.37	0.38	0.38
Energy Appliances Dataset										
OLS	<b>0.0</b>	0.46	0.41	0.36	0.37	0.54	0.51	0.51	0.51	0.51
R0.1	<b>0.0</b>	0.44	0.39	0.35	0.35	0.54	0.51	0.51	0.51	0.51
R0.5	<b>0.0</b>	0.45	0.41	0.37	0.37	0.55	0.52	0.53	0.53	0.53
R0.9	<b>0.0</b>	0.43	0.41	0.38	0.38	0.54	0.51	0.51	0.51	0.51
Superconductivity Dataset										
OLS	<b>0.0</b>	0.07	0.07	0.06	0.06	0.44	0.37	0.37	0.37	0.37
R0.1	<b>0.0</b>	0.07	0.07	0.06	0.06	0.42	0.37	0.37	0.37	0.37
R0.5	<b>0.0</b>	0.04	0.04	0.04	0.04	0.43	0.39	0.39	0.39	0.39
R0.9	<b>0.0</b>	0.04	0.04	0.03	0.03	0.53	0.49	0.49	0.5	0.5

## 5 Results and Discussion

Table 1 shows the median CoSS scores and RMSE scores for CoFE and other feature selection approaches (refer Table S1 and Table S2 in supplementary for details). CoFE has the lowest median CoSS scores as compared to all other approaches indicating consistency. Further, the median RMSE scores for CoFE is higher but comparable to other approaches. The difference in median CoSS scores



**Fig. 2.** ECDF plots for CoSS and RMSE scores of all feature selection approaches for Linear Regression(OLS)



**Fig. 3.** ECDF plots for CoSS and RMSE scores of all feature selection approaches for Ridge Regression model with  $\alpha = 0.5$

is much higher than the difference in RMSE values. This can also be observed in the Empirical Cumulative Distribution Function (ECDF) plots in Fig. 2 and Fig. 3. In Fig. 2 we depict the ECDF plots of the CoSS scores (top row) and the RMSE scores (bottom row) for all the feature selection approaches using Linear

Regression (OLS) model. It can be seen that the ECDF plots of all approaches show similar performance in the RMSE plots while for the ECDF plots for CoSS, CoFE shows significant improvement as compared to other approaches. The same trend can also be seen in the ECDF plots of Fig. 3 for Ridge Regression with  $\alpha$  values of 0.5 (refer Figure S1 and Figure S2 for Ridge Regression with  $\alpha = 0.1$  and  $\alpha = 0.9$ ).

**5.1 CoSS Gain**

To investigate the CoSS gain of CoFE against other approaches, we conduct Mann-Whitney U tests [22] and compute the effect size using Cliff’s Delta [8]. We postulate the null hypothesis  $H_0$  as “There is no difference in the distribution of the CoSS scores of CoFE vs other approaches,” and the alternative hypothesis was  $H_a$  as “The underlying distribution of CoSS scores of CoFE is **lesser** than that of other approaches”.

The results of Mann-Whitney U tests [22] and the effect size (Cliff’s Delta [8]) are provided in Table 2, Table 3 (refer Table S5 and Table S6 for additional results). The column FS denotes the other approach (out of RFE, SFS, SBS and BD) compared against CoFE. While the columns U and p-val indicates the U Statistic and the p-value of the test. We also use Cliff’s Delta to calculate the effect size of all the tests ( $\Delta$ ).

**Table 2.** Mann-Whitney U test for CoSS gain and RMSE loss of CoFE as compared to other techniques for Linear Regression (OLS)

FS	CoSS Gain			RMSE Loss		
	U	p-val	$\Delta$	U	p-val	$\Delta$
Housing Price Dataset						
RFE	3.5e+06	0.0	0.76	2.5e+06	2.9e−37	0.23
SFS	3.7e+06	0.0	0.84	2.8e+06	3.0e−96	0.38
SBS	3.7e+06	0.0	0.83	2.7e+06	7.5e−89	0.36
BD	3.7e+06	0.0	0.83	2.7e+06	3.1e−92	0.37
Energy Appliances Dataset						
RFE	2.9e+05	8.4e−80	0.62	2.0e+05	1.4e−03	0.1
SFS	2.9e+05	5.7e−76	0.6	2.0e+05	1.5e−03	0.1
SBS	2.8e+05	1.4e−63	0.55	2.0e+05	1.8e−03	0.1
BD	2.80e+05	5.3e−65	0.56	2.0e+05	1.8e−03	0.1
Superconductivity Dataset						
RFE	3.5e+05	8.5e−175	0.92	2.3e+05	3.8e−15	0.26
SFS	3.5e+05	1.3e−177	0.93	2.3e+05	3.2e−15	0.26
SBS	3.4e+05	3.2e−165	0.89	2.2e+05	2.0e−13	0.24
BD	3.4e+05	2.8e−165	0.89	2.2e+05	1.5e−13	0.24

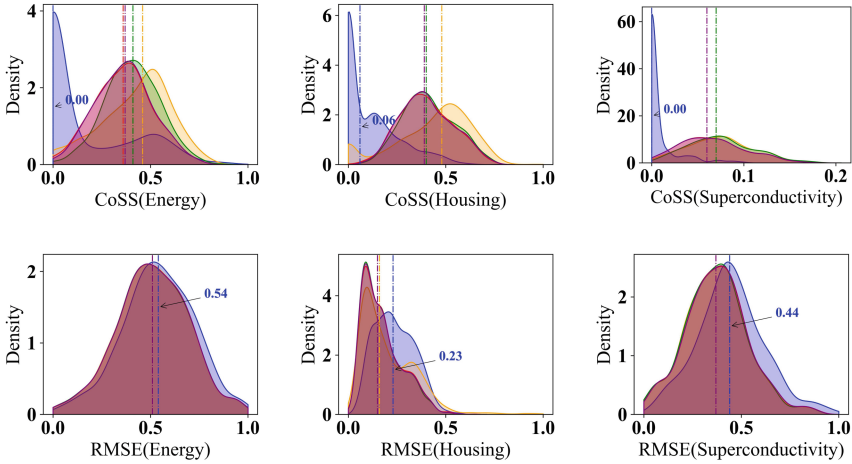
### 5.2 RMSE Loss

Similar to previous analysis, we also investigate the loss in RMSE of proposed CoFE against other feature selection approaches not offering consistency. We postulate the null hypothesis  $H_0$  as “There is no difference in the distribution of the RMSE scores of CoFE vs other approaches” and the alternative hypothesis as  $H_a$  was “The underlying distribution of RMSE scores of CoFE is **greater** than that of other approaches”. We include CoSS gain and RMSE loss as reported in Table 2 and Table 3 (Additional results in Table S5 and Table S6).

We conducted Mann-Whitney U tests to compute statistical significance for our observations regarding CoFE showing lower CoSS score (i.e., CoSS gain) and CoFE showing higher RMSE score (i.e., RMSE loss). Further, the effect size enables us to compare both the CoSS gain and RMSE loss to prove that CoFE has substantial more gain in CoSS than the resulting loss in RMSE. The extremely low p-values and large U statistics for both CoSS gain and the RMSE loss reported in the tables provide robust statistical evidence that CoFE has lower CoSS scores but it has higher RMSE score as compared to other approaches. Further the effect size (Cliff’s Delta) values for the tests in the CoSS Gain section are substantially larger than those of the RMSE Loss section. We further provide the comparison of the effect sizes of all the tests in Fig. 5 for a visual illustration. We use the interpretation of Cliff’s Deltas as <0.15 as negligible, 0.15 to 0.33 as small, 0.33 to 0.45 as medium and above 0.45 as large

**Table 3.** Mann-Whitney U test for CoSS gain and RMSE loss of CoFE as compared to other techniques for Ridge Regression with  $\alpha=0.5$

FS	CoSS Gain			RMSE Loss		
	U	p-val	$\Delta$	U	p-val	$\Delta$
Housing Dataset						
RFE	3.1e+05	7.7e-101	0.71	2.3e+05	1.5e-15	0.26
SFS	3.1e+05	6.6e-112	0.75	2.3e+05	1.2e-14	0.25
SBS	3.1e+05	4.7e-108	0.73	2.2e+05	3.8e-14	0.25
BD	3.1e+05	8.2e-108	0.73	2.2e+05	5.2e-14	0.25
Energy Dataset						
RFE	3.3e+05	6.5e-153	0.85	2.0e+05	2.6e-03	0.09
SFS	3.3e+05	3.2e-149	0.84	2.0e+05	3.2e-03	0.09
SBS	3.2e+05	3.5e-135	0.8	2.0e+05	3.4e-03	0.09
BD	3.3e+05	4.2e-136	0.81	2.0e+05	3.4e-03	0.09
Superconductivity Dataset						
RFE	3.2e+05	5.9e-128	0.79	2.1e+05	5.6e-08	0.18
SFS	3.2e+05	3.1e-125	0.79	2.1e+05	5.3e-08	0.18
SBS	3.1e+05	8.5e-108	0.73	2.1e+05	2.4e-07	0.17
BD	3.1e+05	1.2e-107	0.73	2.1e+05	2.5e-07	0.17



**Fig. 4.** Density plots for CoSS and RMSE scores of all feature selection approaches for OLS

[23]. As seen in the Fig. 5, the Coss gain effect sizes are significantly larger than that of RMSE loss showing the minimal loss in RMSE for CoFE compared to the stability gain achieved.

### 5.3 Jensen-Shannon Distance of CoSS and RMSE Scores

We compare the distributions of CoSS Scores and the RMSE scores of CoFE with all the approaches (i.e. RFE, SFS, SBS and BD) for both Linear Regression (OLS) and Ridge Regression models for all datasets using Jensen-Shannon Distance (JSD) [21] in Table 4. The column ‘M’ denotes the model and the sections CoSS and RMSE depicts the JSD of CoSS scores of CoFE vs the same for other approaches. As seen from the table, the distribution of CoSS scores (CoFE vs other approaches) are substantially different owing to the high JSD values. The same can be visualized in the density plots of Fig. 4 where the CoSS distribution of CoFE is substantially different from that of other approaches, while the distribution of RMSE is similar (refer Figure S3, Figure S4, and Figure S5 for all density plots). The JSD values for the distribution of RMSE scores for CoFE and other approaches are relatively low indicating that CoFE can achieve substantial boost in consistency of coefficient’s sign (CoSS scores) with low loss of accuracy (RMSE) as compared to other approaches for feature selection.

### 5.4 Discussion

The results from the previous sub-section provides robust statistical evidence to indicate that models built on features selected using CoFE are substantially consistent as compared to models built using other approaches with a considerably low loss in accuracy (RMSE). This makes it more suitable for XAI where the coefficients of the Linear Model are interpreted to explain the decision.

## 6 Visualizing CoFE’s Impact on Explainability

We plot the coefficients from a random run using the Housing Dataset to visualize the impact of CoFE on the variability of coefficients’ sign in Fig. 6. The OLS

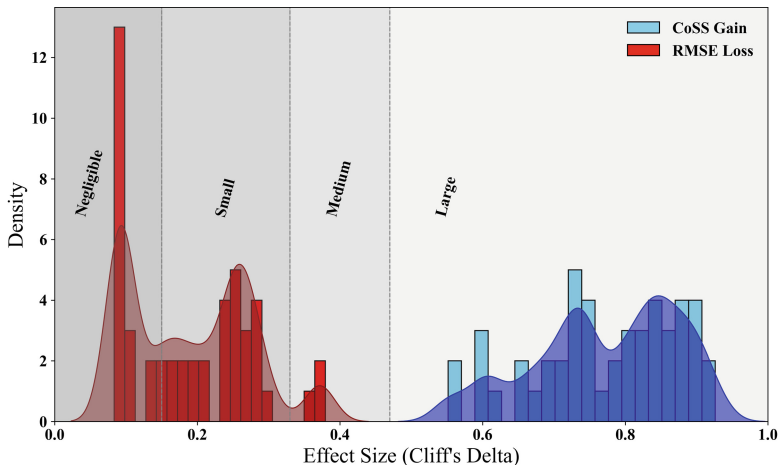
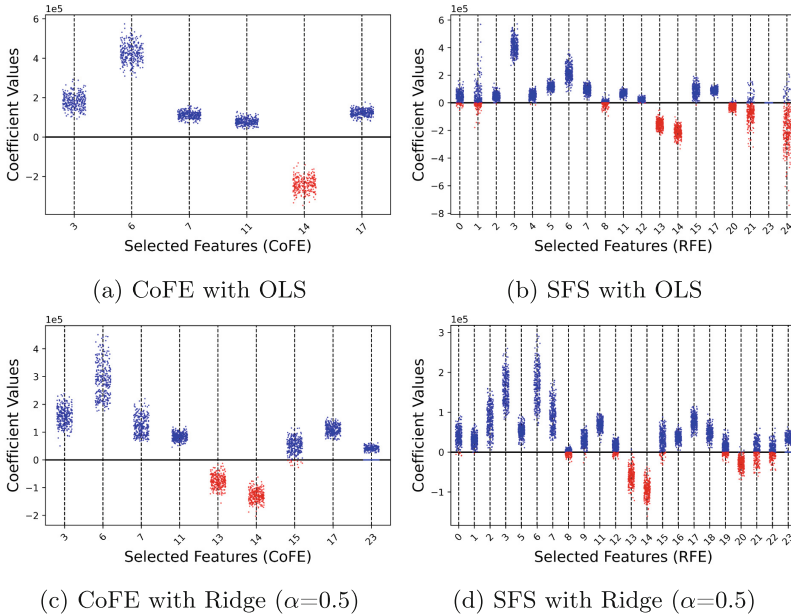


Fig. 5. Plot of Effect Size

Table 4. Jensen-Shannon Distance of the CoSS distribution of CoFE with the CoSS distribution of other techniques

M	CoSS				RMSE			
	RFE	SFS	SBS	BD	RFE	SFS	SBS	BD
Housing Price Dataset								
OLS	0.529	0.599	0.583	0.59	0.224	0.246	0.24	0.242
R0.1	0.548	0.591	0.59	0.588	0.181	0.175	0.173	0.173
R0.5	0.467	0.513	0.502	0.502	0.179	0.167	0.167	0.166
R0.9	0.458	0.513	0.507	0.516	0.195	0.187	0.177	0.181
Energy Appliances Dataset								
OLS	0.478	0.546	0.519	0.53	0.086	0.078	0.076	0.076
R0.1	0.539	0.565	0.553	0.553	0.088	0.081	0.08	0.078
R0.5	0.622	0.634	0.611	0.614	0.071	0.072	0.069	0.069
R0.9	0.651	0.653	0.629	0.63	0.075	0.069	0.068	0.067
Superconductivity Dataset								
OLS	0.642	0.661	0.618	0.618	0.176	0.176	0.169	0.169
R0.1	0.637	0.648	0.591	0.591	0.145	0.144	0.136	0.138
R0.5	0.53	0.514	0.452	0.451	0.112	0.113	0.106	0.106
R0.9	0.444	0.43	0.376	0.382	0.102	0.105	0.098	0.099

models were trained on features selected by CoFE (in Fig. 6a) and RFE (in Fig. 6b). The Ridge Regression model (with  $\alpha = 0.5$ ) was trained on features by CoFE (in Fig. 6c) and RFE (in Fig. 6d). Blue color indicates a positive value while red color indicates a negative value of the coefficients. The coefficients of the models trained on the features selected by CoFE shows no/very low variability in their coefficients' sign. However, the models trained on the features selected using RFE shows high variability of coefficients' sign making it inconsistent for explainability tasks (refer Section S5 for visualization with other feature selection approaches).



**Fig. 6.** Plot for impact on consistency of selected coefficients in Housing dataset with coefficient values from 300 random splits. The CoSS value for CoFE was 0.0 and for RFE it was 0.40 with OLS and for Ridge ( $\alpha = 0.5$ ) the CoSS value for CoFE was 0.10 and for RFE it was 0.37). Blue color indicates positive values of the coefficients while red color indicates negative values of the coefficients. (Color figure online)

## 7 Conclusion

Linear regression model with OLS and Ridge regression, trained on selected features from traditional approaches, exhibited low consistency of coefficients' sign. In the context of Explainable Artificial Intelligence (XAI), the consistency of features for inherent data variability is crucial. For applications valuing interpretability, relying solely on traditional feature selection might not fully reveal

the directional impacts of features. Our proposed approach for feature selection demonstrated substantial improvement in the consistency of coefficients' signs while maintaining comparable accuracy to existing approaches. Thus, balancing feature consistency with predictive performance is essential for developing interpretable and reliable linear regression models. We recommend further research in this area to expand feature selection approaches across various model types from an XAI perspective.

## References

1. Agarwal, C., et al.: Rethinking stability for attribution-based explanations. arXiv preprint [arXiv:2203.06877](https://arxiv.org/abs/2203.06877) (2022)
2. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods. arXiv preprint [arXiv:1806.08049](https://arxiv.org/abs/1806.08049) (2018)
3. Díaz-Rodríguez, N., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020)
4. Borgonovo, E., Plischke, E.: Sensitivity analysis: a review of recent advances. *Eur. J. Oper. Res.* **248**(3), 869–887 (2016)
5. Candanedo, L.M., Feldheim, V., Deramaix, D.: Data driven prediction models of energy use of appliances in a low-energy house. *Energy Build.* **140**, 81–97 (2017)
6. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Comput. Electr. Eng.* **40**(1), 16–28 (2014)
7. Chatterjee, S., Hadi, A.S.: *Sensitivity Analysis in Linear Regression*. Wiley (2009)
8. Cliff, N.: Dominance statistics: ordinal analyses to answer ordinal questions. *Psychol. Bull.* **114**(3), 494 (1993)
9. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: *Least angle regression* (2004)
10. Ferri, F.J., Pudil, P., Hatef, M., Kittler, J.: Comparative study of techniques for large-scale feature selection. In: *Machine Intelligence and Pattern Recognition*, vol. 16, pp. 403–413. Elsevier (1994)
11. Fox, J., Weisberg, S.: *An R Companion to Applied Regression*. Sage publications (2018)
12. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**(Mar), 1157–1182 (2003)
13. Hamidieh, K.: A data-driven statistical model for predicting the critical temperature of a superconductor. *Comput. Mater. Sci.* **154**, 346–354 (2018)
14. Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12**(1), 55–67 (1970)
15. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Dellino, G., Meloni, C. (eds.) *Uncertainty Management in Simulation-Optimization of Complex Systems*. ORSIS, vol. 59, pp. 101–122. Springer, Boston, MA (2015). [https://doi.org/10.1007/978-1-4899-7547-8\\_5](https://doi.org/10.1007/978-1-4899-7547-8_5)
16. Islam, M.R., Lima, A.A., Das, S.C., Mridha, M.F., Prodeep, A.R., Watanobe, Y.: A comprehensive survey on the process, methods, evaluation, and challenges of feature selection. *IEEE Access* **10**, 99595–99632 (2022)
17. Jacovi, A., Marasović, A., Miller, T., Goldberg, Y.: Formalizing trust in artificial intelligence: prerequisites, causes and goals of human trust in AI. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 624–635 (2021)



18. Kaggle: house prices: advanced regression techniques (2024). <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data> Accessed 6 April 2024
19. Khalid, S., Khalil, T., Nasreen, S.: A survey of feature selection and feature extraction techniques in machine learning. In: 2014 Science and Information Conference, pp. 372–378. IEEE (2014)
20. Kittler, J.: Feature set search algorithms. In: Pattern recognition and signal processing (1978)
21. Lin, J.: Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **37**(1), 145–151 (1991)
22. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* **18**, 50–60 (1947)
23. Meissel, K., Yao, E.S.: Using cliff’s delta as a non-parametric effect size measure: an accessible web app and r tutorial. *Pract. Assess. Res. Eval.* **29**(1) (2024)
24. Mitchell, T.M.: *Machine learning* (1997)
25. Pudil, P., Novovičová, J., Bláha, S.: Statistical approach to pattern recognition: theory and practical solution by means of preditas system. *Kybernetika* **27**(7), 1–3 (1991)
26. Reunanen, J.: Overfitting in making comparisons between variable selection methods. *J. Mach. Learn. Res.* **3**(Mar), 1371–1382 (2003)
27. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?” Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
28. Rudin, C.: Please stop explaining black box models for high stakes decisions. *Stat* **1050**(26), 457 (2018)
29. Scott, D.W.: *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley (2015)
30. Shankaranarayana, S.M., Runje, D.: ALIME: autoencoder based approach for local interpretability. In: Yin, H., Camacho, D., Tino, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R. (eds.) *IDEAL 2019. LNCS*, vol. 11871, pp. 454–463. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-33607-3\\_49](https://doi.org/10.1007/978-3-030-33607-3_49)
31. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat Methodol.* **58**(1), 267–288 (1996)
32. Zafar, M.R., Khan, N.M.: DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. arXiv preprint [arXiv:1906.10263](https://arxiv.org/abs/1906.10263) (2019)
33. Zhao, X., Huang, W., Huang, X., Robu, V., Flynn, D.: BayLIME: Bayesian local interpretable model-agnostic explanations. In: *Uncertainty in Artificial Intelligence*, pp. 887–896. PMLR (2021)
34. Zhou, Z., Hooker, G., Wang, F.: S-lime: Stabilized-lime for model explanation. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2429–2438 (2021)



# From One to Many Lorikeets: Discovering Image Analogies in the CLIP Space

Songlong Xing<sup>1</sup>(✉), Elia Peruzzo<sup>1</sup>, Enver Sangineto<sup>2</sup>, and Nicu Sebe<sup>1</sup>

<sup>1</sup> University of Trento, Trento, Italy  
songlong.xing@unitn.it

<sup>2</sup> University of Modena and Reggio Emilia, Modena, Italy

**Abstract.** Drawing analogies between two pairs of entities in the form of  $A:B::C:D$  (i.e. A is to B as C is to D) is a hallmark of human intelligence, as evidenced by sufficient findings in cognitive science for the last decades. In recent years, this property has been found far beyond cognitive science. Notable examples are `word2vec` and `GloVe` models in natural language processing. Recent research in computer vision also found the property of analogies in the feature space of a pretrained ConvNet feature extractor. However, analogy mining in the semantic space of recent strong foundation models such as CLIP is still understudied, despite the fact that they have been successfully applied to a wide range of downstream tasks. In this work, we show that CLIP possesses the similar ability of analogical reasoning in the latent space, and propose a novel strategy to extract analogies between pairs of images in the CLIP space. We compute all the difference vectors of a pair of any two images that belong to the same class in the CLIP space, and employ k-means clustering to group the difference vectors into clusters irrespective of their classes. This procedure results in cluster centroids representative of class-agnostic semantic analogies between images. Through extensive analysis, we show that the property of drawing analogies between images also exists in the CLIP space, which are interpretable by humans through a visualisation of the learned clusters.

**Keywords:** representation learning · foundation models · latent space understanding

## 1 Introduction

The ability of drawing analogies is a key characteristic of human intelligence [34]. For decades, this property has been well-studied in the field of cognitive science. A classical model, known as the ‘parallelogram model’ [35], describes the relations between a pair of entities in the vector space, assuming that the difference vectors of two pairs of entities that reflect the same analogy should be

---

**Supplementary Information** The online version contains supplementary material available at [https://doi.org/10.1007/978-3-031-78189-6\\_25](https://doi.org/10.1007/978-3-031-78189-6_25).

close in the vector space [30]. This model formulates a  $A:B::C:?$  question (*i.e.*,  $A$  is to  $B$  as  $C$  is to  $?$ ) for analogical reasoning, which is still important in testing the reasoning abilities of humans and AI. For instance, given a pair of images of a dog indoors ( $A$ ) and a counterpart in the wild ( $B$ ), it is easy for a child to pick an image of a cat in the wild ( $D$ ) from a bunch of candidates after being shown a cat indoors ( $C$ ), to match the underlying relation between the dog images. Analogies allow generalization because one can transfer existing knowledge from one context to another [14].

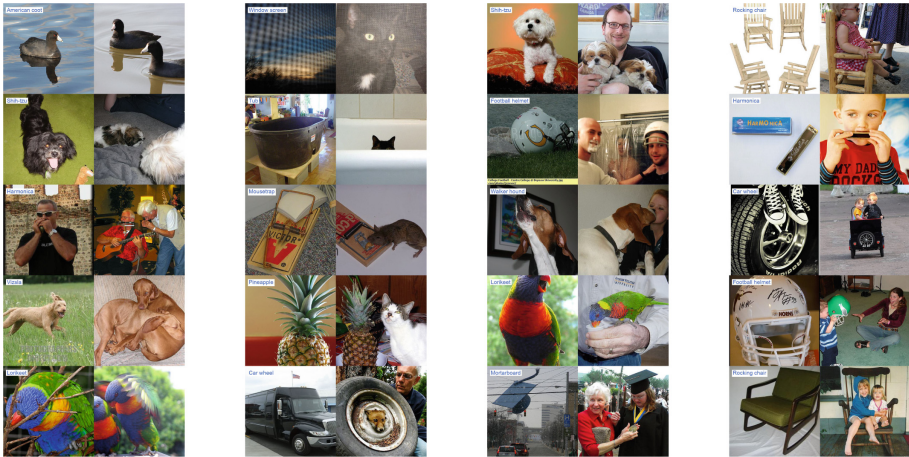
In the past years, the property of reasoning by analogy has been found far beyond the field of cognitive science. In the Natural Language Processing (NLP) community, notable examples include `word2vec` [27] and `GloVe` [29]. These models exhibit the ability to draw basic analogies in the word embedding space that conform to the ‘parallelogram law’, *e.g.*,  $w_{queen} - w_{king} + w_{man} \approx w_{woman}$ , where  $w_x$  is a word embedding for the word  $x$  [12, 27]. Recently, Transformer-based language models such as GPT-3 [7] are found to be capable of answering more sophisticated analogy questions [38]. Drawing analogies between images has also attracted research interests in the computer vision community [6, 18, 33]. In this paper, we define an ‘analogy’ as a class-agnostic semantic change that generalizes to multiple classes. For instance, an analogy describing the numerical change of ‘one-to-many’ applies to most tangible concepts (*e.g.*, real-life objects). Examples are provided in Fig. 1.

Unlike many well-known language models such as `word2vec` [27], exploring the property of analogy making in the feature space of visual models to understand how they represent general semantic changes across images remains an inspiring yet understudied topic. Recently, Hariharan and Girshick [17] proposed to explore analogies in the feature space of a ConvNet feature extractor, and trained a generative model to hallucinate features in few-shot learning scenarios. The way they exploit analogies between image representations is also based on an implicit assumption of the parallelogram law. Recently, large-scale multimodal foundation models trained with a contrastive loss gain popularity [23, 31], with CLIP [31] being an important representative. These models have been applied to a wide range of downstream tasks successfully in a zero-shot manner and exhibited impressive performance. However, to our best knowledge, no existing research explores the property of analogy-making in the latent space of these foundation models. Does CLIP possess the similar ability of drawing analogies in the shared space as found in previous language models [27, 29] and vision models [17]? We investigate this question and give a positive answer.

To discover analogies between images, we follow the parallelogram assumption that two pairs of images which reflect the same semantic change should correspond to two difference vectors that are close to each other in the CLIP space. This assumption is intuitive and renders it easy for manipulation. Note



**Fig. 1.** An illustration of an analogy discovered in the CLIP space with our proposed strategy, described in the form of  $A:B::C:D$ .



(a) An analogy that reflects a semantic (numerical) shift of ‘one object to many objects’.

(b) An analogy that reflects a semantic shift of ‘an object to an object with an animal’.

(c) An analogy that reflects a semantic shift of ‘an object to an object with a human adult’.

(d) An analogy that reflects the semantic shift of ‘an object to an object with a human child’.

**Fig. 2.** Examples of four clusters discovered from the CLIP space, each visualised by a subfigure and representing a class-agnostic analogy. Each row contains a pair of images belonging to the same class whose difference vector is assigned to a cluster. The textual name of the class of each image pair is written on the top-left corner of the first image. All the images are cropped to square sizes for better display. Better zoomed in and viewed in colour

that although CLIP is equipped with two encoders for vision and language, respectively, we focus on exploring analogies based on images. We believe that similar findings can be drawn for the textual modality and leave it for future work. We first employ the frozen vision encoder to encode each image in the training set of ImageNet [9] into a vectorial representation in the CLIP space. Within each class, we compute the difference vectors of any two images. Each difference vector is represented as a point in a vector space with the same dimensionality of the CLIP space. We then employ *k-means* clustering [22] over the difference vectors to group them into a fixed number of clusters, irrespective of their classes. Through visualization over the learned clusters, we find that most clusters represent a certain class-agnostic semantic change interpretable to humans, which indicates that the CLIP space possesses the similar property of analogy-making that follows the ‘parallelogram law’, as found in the embedding space (or feature space) of previous models. Some examples are provided in Fig. 2. Our contributions are summarised as follows:

- We show that the CLIP space possesses the property of analogy making as found in the embedding space of well-established word models (*e.g.*, word2vec) [27] and the feature space of ConvNet [17]. To our best knowl-

edge, no existing research has studied the ability of analogical reasoning in the CLIP space.

- Following the parallelogram assumption, we propose a clustering-based strategy to discover analogies in the CLIP space. Visualization shows that the learned clusters represent semantic changes interpretable to humans. We hope that our work will inspire research in the understanding of image semantics encoded in the CLIP space.

## 2 Related Work

Closely related to our work are two streams of research, *i.e.*, (1) analogical reasoning in the latent space of existing neural models, and (2) the interpretation and manipulation of the CLIP space.

**Analogy.** Analogical reasoning refers to the ability of recognising similar patterns among two or more sets of entities. This ability has been studied extensively in the field of cognitive science [20]. Rumelhart *et al.* proposed the classical parallelogram model [35], which assumes that two pairs of entities reflecting the same semantic change translate to two difference vectors that are close in the vector space. This assumption is found to hold in the embedding space of language models that have been trained on large corpuses of text, such as `word2vec` [27] and `GloVe` [29]. There is also work that attempts to understand the analogical abilities of word embeddings [2, 3, 12, 15]. Ushio *et al.* analyse the analogical abilities of Transformer-based language models [38]. Hertzmann *et al.* [18] first introduced the concept of *Image Analogy*, and proposed a statistical approach to transform an image  $B$  to  $B'$  given a pair of images  $A$  and  $A'$  as reference. However, their proposed concept of *image analogy* differs from the *analogy* studied in this paper. Specifically, they focus on pixel-level transformations between images, *e.g.*, change in image resolution, while *analogies* in our work refer to general changes in semantics. Recently, Bar *et al.* [5] propose to construct a grid-like image concatenated with a pair of input and output images from the downstream task and a query input image, and prompt a pretrained image inpainting model to generate the desired image output without further finetuning, a process which they term *visual prompting*. This work is still based on low-level *analogy* between images because image semantics is not considered. Šubrtová *et al.* [37] propose to employ diffusion models [19] to edit an image in higher-level semantics, given a pair of images which specifies the desired transition. Compared to previous studies, [37] focuses on high-level semantic transformations across images. However, their goal is to prompt the generative model to transform the query image based on the intended semantic change manifested by an exemplar image pair. In comparison, our goal is to investigate the ability of CLIP to draw class-agnostic analogies, and propose a strategy to discover such analogies in the CLIP latent space from a large pool of images without relying on any exemplars. Understanding the analogies between images in the feature space is much less studied. Some research efforts employ vector arithmetic for face image manipulation [32]. Hariharan and Girshick [17] proposed to extract analogies from the feature space

of a ConvNet feature extractor. They first train the feature extractor with base classes, and group the feature vectors within each class into clusters. Then they search for two pairs of clusters from two classes such that  $c_1^a - c_2^a$  and  $c_1^b - c_2^b$  (with  $c_i^j$  denoting the  $i$ -th cluster of class  $j$ ) have positive cosine similarity, and collect these quadruplets  $(c_1^a, c_2^a, c_1^b, c_2^b)$  as a dataset, which is used to train a generative model that outputs the synthesized feature  $c_2^b$  given  $[c_1^a, c_2^a, c_1^b]$  as input. This work is closest to ours in the idea of exploring analogies across images in the latent space of a well-trained model. However, the motivation in their work is to hallucinate features in scenarios where training data are scarce, whereas we aim to investigate in the CLIP space [31] the property of semantic analogies as found in previous models [27, 29] and propose a strategy to extract such class-agnostic analogies. In addition, they perform clustering over the feature vectors in each class, while we collect difference vectors of a pair of any two images belonging to the same class, and perform clustering over these difference vectors irrespective of classes. To sum up, this paper is vastly different from previous studies as the initial exploration in the latent space of a foundation model to investigate its ability to draw analogies in terms of image semantics. We also propose an effective clustering-based strategy to explicitly extract such general analogies without relying on exemplars or generative models.

**CLIP Space Interpretations.** With the increasing popularity and use of CLIP [31], there has been some research work contributing to understanding representations in the CLIP space [16, 24, 26]. Recently, Gandelsman *et al.* propose to decompose the image representations as summands and interpret them with textual representations [13]. To our best knowledge, no existing work aims to discover analogies between images in the CLIP space.

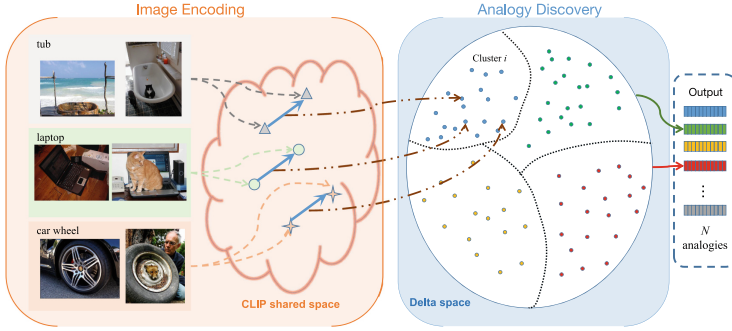
### 3 Method

In this section, we explore the CLIP space [31] to investigate whether it has the ability of analogy-making over image pairs using their image encodings. We also propose a simple yet effective strategy to discover these class-agnostic analogies that reflect general semantic changes.

We make the following assumptions: (i) the analogies in the semantics of images translate to some vector arithmetic operations with their embeddings in the latent space of a well-trained model (CLIP in our case); (ii) provided that  $A:A'::B:B'$  holds, it holds that  $E_v(A) - E_v(A') \approx E_v(B) - E_v(B')$ , where  $E_v(X)$  denotes the image encoding of image  $X$  in the CLIP space (the parallelogram assumption). The overview of the proposed strategy is provided in Fig. 3.

#### 3.1 Image Encoding

To discover the analogies between images, we choose ImageNet [9] as our base dataset because it covers a large number of classes, each containing  $\sim 1,000$  image samples, which spans a large analogy space. Suppose that we have  $K$  classes in the dataset (which is 1,000 for ImageNet), for each class  $c_i, i =$



**Fig. 3.** The diagram of our proposed analogy discovery strategy in the CLIP space. In the **Image Encoding** phase, each image is encoded into a vector representation in the CLIP space with the frozen vision encoder. Different classes are denoted with different shapes. The difference vector of a pair of images is represented as a point in the *delta space*, where **Analogy Discovery** is performed with off-the-shelf clustering algorithms.

$1, 2, \dots, K$ , there are  $n_i$  raw images. We employ the CLIP vision encoder to encode all  $n_i$  images corresponding to class  $c_i$  into the CLIP space:

$$x_j^i = E(X_j^i), \text{ for } j \in \{1, 2, \dots, n_i\} \tag{1}$$

where  $x_j^i \in \mathcal{R}^d$  is the image encoding of image  $X_j^i$ , and  $d$  is the dimensionality of the shared space of CLIP.

After this process, we have  $K$  sets of image encodings  $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_K$ , where  $\mathcal{I}_i = \{x_1^i, x_2^i, \dots, x_{n_i}^i\}$  is the set of image encodings for class  $c_i$ , with each  $x_j^i$  corresponding to a point in the CLIP space.

### 3.2 Analogy Discovery

We aim to discover general analogies irrespective of classes. For each class  $c_i$  with a set of  $n_i$  images,  $n_i \times (n_i - 1)$  image pairs can be derived from their encodings  $\mathcal{I}_i$ , each reflecting a semantic change. Note that an image pair, denoted as  $A:B$ , is asymmetric because  $B:A$  would reflect a reverse semantic shift.

To extract class-agnostic analogies, we collect each possible image pair belonging to all classes into a pool:

$$\mathcal{Q} = \{x_j^i - x_k^i \mid j, k \in [1, n_i], j \neq k, i \in [1, 2, \dots, K]\} \tag{2}$$

Each sample in  $\mathcal{Q}$  is associated with a difference vector in the CLIP space. We define a vector space termed as the *delta space* that has the same dimensionality with the CLIP space:

$$S^\Delta = \{v_x - v_y \mid v_x, v_y \in S^{CLIP}\} \tag{3}$$



In this way, each difference vector in  $\mathcal{Q}$  can be represented as a data point in  $S^\Delta$ . According to the parallelogram assumption, a class-agnostic analogy exemplified by  $\mathbf{A}:\mathbf{A}'::\mathbf{B}:\mathbf{B}'::\dots$  can be represented with  $E_v(A) - E_v(A') \approx E_v(B) - E_v(B')$ . However, one notable difference between CLIP and other backbone networks [17, 27, 29] is that it is pretrained with a contrastive loss based on text-image cosine similarity. Therefore, we make a slight modification to the conventional parallelogram assumption (which is based on Euclidean distance) and employ cosine similarity as the metric to measure the distance between any two points in  $S^\Delta$ , instead of Euclidean distance as in previous models [27, 29]. In this sense, provided that  $\mathbf{A}:\mathbf{A}'::\mathbf{B}:\mathbf{B}'$  holds, the parallelogram assumption can be rewritten as follows:

$$\left\langle \frac{E_v(A) - E_v(A')}{\|E_v(A) - E_v(A')\|}, \frac{E_v(B) - E_v(B')}{\|E_v(B) - E_v(B')\|} \right\rangle \approx 1 \quad (4)$$

To discover these analogies, we group  $\mathcal{Q}$  into clusters in  $S^\Delta$ , as illustrated in Fig. 3. A straightforward solution is to employ *k-means* clustering [22] over all the data points in  $\mathcal{Q}$ , which we find to extract meaningful analogies on smaller datasets *e.g.*, ImageNet-100 [9]. However, there remain two issues that need to be addressed: (i) *k-means* clustering is performed based on Euclidean distance; (ii) when we aim to extract analogies on large-scale datasets, employing *k-means* naively to iterate over all points in  $\mathcal{Q}$  would be impractical because the number of points is prohibitively large ( $\sim 10^9$  for ImageNet). We address these two issues as follows:

(i) We show that partitioning a set of data points into clusters in a vector space by their cosine similarity is equivalent to first  $L_2$ -normalising all data points into a  $d$ -sphere, and then employing regular clustering based on Euclidean distance:

$$\begin{aligned} \|x - y\|^2 &= \|x\|^2 + \|y\|^2 - 2\langle x, y \rangle \\ &= 2(1 - \cos \angle(x, y)) \end{aligned} \quad (5)$$

where  $\langle \cdot, \cdot \rangle$  denotes inner product.

Therefore, to perform clustering based on cosine similarity, we apply  $L_2$  normalisation before employing regular *k-means* clustering over the data points  $\mathcal{Q}$  in space  $S^\Delta$ .

(ii) To avoid iterating over all data points in  $\mathcal{Q}$ , we employ the minibatch version of *k-means* clustering [36], which samples a batch of data points and updates the cluster centroids in each iteration. We provide more details on minibatch *k-means* clustering and the method we use to efficiently sample a batch of difference vectors of image pairs on the fly in Appendix.

After the clustering process, we get  $N$  clusters representative of  $N$  class-agnostic analogies. We use the cluster centroids to represent the analogies:

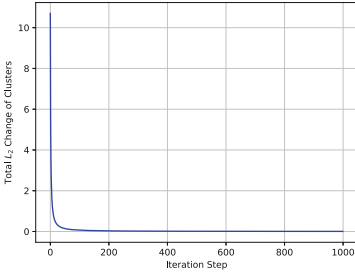
$$\mathbf{a}_l = \frac{1}{|A_l|} \sum_{\mathbf{v} \in A_l} \mathbf{v}, \quad l \in [1, 2, \dots, N] \quad (6)$$

where  $A_l$  is the set of data points that are assigned to the  $l$ -th cluster. For minibatch *k-means*,  $\mathbf{a}$  is computed on the fly without keeping the data points.



## 4 Experiments

### 4.1 Experimental Details



**Fig. 4.** Total change of all clusters at each iteration in terms of  $L_2$  distance. The value at the  $t$ -th iteration is computed as  $\sum_l \| \mathbf{a}_{l,t} - \mathbf{a}_{l,t-1} \|$ , with  $\mathbf{a}_{l,t}$  being the  $l$ -th cluster centroid at the current iteration

day to encode all image samples. We save the image representations for the following phase. In the phase of **Analogy Discovery**, we employ the off-the-shelf minibatch *k-means* clustering algorithm implemented by the scikit-learn library [28], which does not require any GPU device and runs only on CPU. We set the number of clusters  $N$  to 256 and keep it fixed. The batch size of clustering is 1,048,576 and we run the clustering algorithm for 1,000 iterations, which takes about 15.5 h. We set the reassignment ratio to 0.1, so that when a cluster contains an overly small number of data points, it can be reassigned. This prevents clusters from learning over-specific analogies which can be exemplified by very few image pairs. The total change of clusters at each iteration (in terms of  $L_2$  distance) decreases sharply at early iterations and decreases smoothly afterwards (Fig. 4), which shows that the proposed clustering-based strategy is convergent. The cluster centroids gradually stabilise with more iterations. In practice, we find that the clusters are able to capture meaningful analogies when the total change of clusters in terms of  $L_2$  distance reaches 0.05 or below. In our experiments, this figure reaches  $5.2e-3$  after 1,000 iterations. To enable the use of GPU devices, we also implement an Exponential Moving Average (EMA) based clustering module with Pytorch which shares the same idea of the mini-batch version of *k-means* clustering. We provide details for this type of implementation in Appendix.

We explore image analogies in the CLIP space<sup>1</sup> based on ImageNet [9], which contains 1,000 image classes. Each class has no more than 1,300 images in the training set. On average, each class has 1,281.167 image samples. Over 1.6 billion possible image pairs can be derived from this dataset, which span a large pool of semantic shifts where analogies can be discovered. Unless otherwise stated, we employ the pretrained ViT-B/16 [10] as the CLIP vision encoder, which we keep frozen throughout the experiments. In the phase of **Image Encoding**, we encode the whole dataset into the CLIP latent space, using one *Quadro RTX 5000* device, which takes approximately one GPU

<sup>1</sup> The code is available at <https://github.com/Sxing2/CLIP-Analogy>.

**Table 1.** Categorisation of analogies observed through our visualisation.

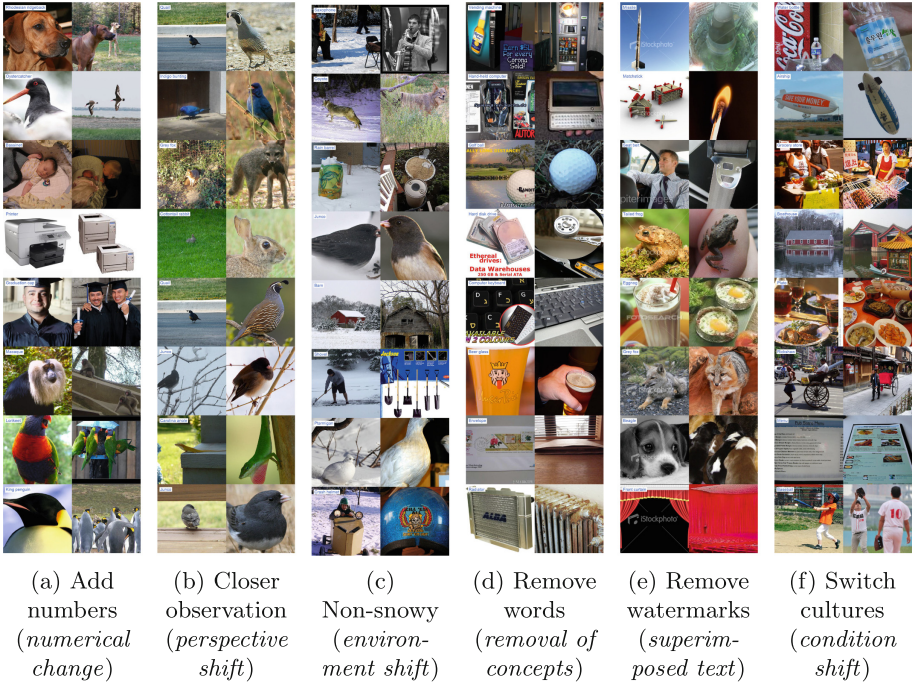
Broad Categories	Types	Definitions
<b>Content-related Analogies</b>	numerical change	<i>change in the numbers of the objects present in the image</i>
	addition/removal of concepts	<i>addition or removal of objects external to the class of interest</i>
	perspective shift	<i>a different perspective view of the object</i>
	environment shift	<i>a different environment of the object</i>
	condition shift	<i>change in the the object condition (e.g.from old to new)</i>
<b>Structure-related Analogies</b>	superimposed text	<i>irrelevant text superimposed onto the image (e.g.watermarks)</i>
	background change	<i>change of background of the image</i>

## 4.2 Discussion

In this section, we qualitatively analyse the analogies extracted with our proposed strategy, and show that the CLIP space exhibits the ability of analogical reasoning. We also show that our clustering-based strategy extracts meaningful image analogies, which are interpretable by humans through a visualisation into the clusters.

**Visualization of Clusters.** We sample a large number (131,072) of image pairs (each belonging to the same class), compute their difference vectors and assign them to the nearest clusters (in terms of cosine similarity) in the  $S^\Delta$  space. We visualize each cluster by sampling and displaying the image pairs assigned to it. These analogies are extracted in the latent space without any supervision, and we observe that they cover a wide range of changes in different semantics levels. In general, these analogies fall into two broad categories, which are *content-related* and *structure-related*. *Content-related* analogies are those semantic changes that lead to a high level of change in the image contents, e.g., the number of objects present. *Structure-related* analogies are related to the change of the relatively low-level structure of an image without affecting the high-level semantics. We summarise the categorisation of the extracted analogies in Table 1. A user survey with 8 human participants is conducted based on this categorisation. We provide 24 questions, each containing five image pairs from a cluster. Both the clusters and the image pairs are randomly sampled without manual picking. The participants are given the definitions of each type of analogies and their examples, and are asked to select from the type of analogies identified in Table 1 that the five image pairs reflect. We also provide a choice of ‘None of the Above’ in case the image pairs do not present any interpretable analogy to the user. For each participant, we shuffle the order of the questions and ask the participant to answer a portion of questions in the survey (instead of setting all the questions as mandatory). A total of 154 answers are collected, and only 11 of them (7.14%) are ‘None of the Above’. This shows that most of the clusters are representative of changes in image semantics interpretable to humans. The proportion of the selected analogy types is provided in Fig. 6a. We give more details and question samples in Appendix.

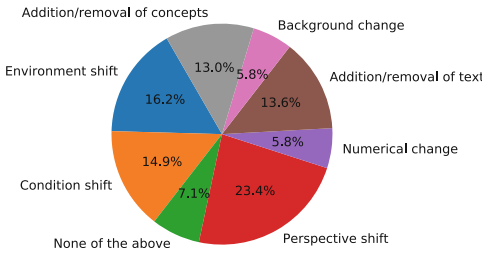
Note that each type can include more than one specific analogy, and that if an analogy is discovered, usually its reverse analogy is discovered by another



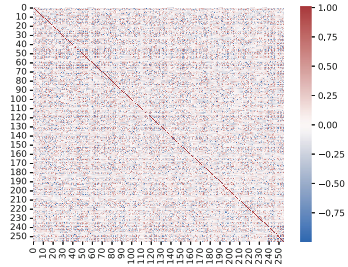
**Fig. 5.** Visualizations of some representative analogies extracted from the CLIP space, each captured by a cluster. For each cluster we show eight image pairs whose difference vectors are assigned to it. The class name of each image pair is written on the top-left corner of the first image. All the images are cropped to square sizes for better display. Better zoomed in and viewed in colour.

cluster, as shown in Appendix. For instance, the type of ‘addition/removal of concepts’ can include the analogies of adding a human adult, a human child or an animal, etc., or removing these concepts. Apart from the results shown in Fig. 2 and Fig. 5, we display more such examples in Appendix for limited space. The fact that the learned clusters are interpretable to humans shows that the CLIP space possesses the ability of making analogies by vector arithmetic operations on the image encodings, a property also found in well-established models in the NLP community [2, 12, 27, 29].

Among the content-related analogies discovered in the clusters, we also observe analogies that require sophisticated reasoning. One such example is given in Fig. 5f, where the cluster learns an analogy that transitions the cultural context of the image. As can be seen in the first, second and seventh rows (image pairs), the text on the object of interest in the source image (on the left) is written in Latin letters, while it is changed into Asian scripts in the target image (on the right). In the third image pair, both the text and the humans present are changed into Asian, while the scene (street vendors looking after a grocery store) is retained across two images. The image pairs in the sixth and



(a) Percentage of analogy types in the user survey.



(b) Pairwise cosine similarity heatmap of all clusters.

**Fig. 6.** Statistics of the learned clusters.

eighth rows can also be explained the same way. In the fourth and fifth pairs, the objects present in the source images are changed into their counterparts with recognisable attributes in an Asian context. This indicates that high-level semantics such as cultural context shifts is encoded by a subspace determined by this extracted analogy vector in the CLIP space. Another interesting example that we observe is the analogy of transitioning source images that contain written words to clean text-free images. As can be seen in Fig. 5d, in the image pairs sampled from the cluster, written words are clearly present in the source images, while target images are free of visual text, with other semantic information in the image unchanged (the object of the class of interest). The disentanglement of written and visual concepts in the CLIP space [24, 26] is an important research topic, as CLIP is shown to be sensitive to typographic attacks [4, 16]. This extracted analogy implies that a hyperplane defined by this analogy vector may be helpful in separating written and visual concepts. We also observe that the CLIP space is able to differentiate written words that are part of the image itself and faint written text superimposed onto the image such as watermarks, as our proposed strategy discovers two clusters that deal with them separately (Fig. 5d and Fig. 5e). We discuss in detail more properties, *e.g.*, magnitude, of the analogies in Appendix.

### 4.3 Similarity of Discovered Analogies

In our experiments, we discover  $N = 256$  clusters which ideally represent an image analogy in the CLIP space. Intuitively, some analogies can reflect more similar semantic changes. For example, analogies of ‘*adding a human adult*’ and ‘*adding a human child*’ are semantically similar, while some can be opposite if they reflect reverse analogies. To analyse how the learned analogies relate to each other, we compute their pairwise cosine similarity, as shown in Fig. 6b. It can be seen that most analogies are almost orthogonal to each other, as reflected by the light colour areas. Some analogies have positive or negative cosine similarity with a larger magnitude. In Appendix, we visualize some of these analogies and

show that these clusters indeed reflect semantically similar or reverse analogies, respectively.

#### 4.4 Effect of Number of Clusters

We investigate the effect of the number of clusters ( $N$ ), which is the most crucial influencing factor in this work. We experiment with 32, 64, 128, 256, and 512 clusters, and report the average cosine similarity of 10 pairs of learned analogies with the highest similarity, and 10 pairs of the lowest. The results are reported in Fig. 7. It can be seen that as  $N$  increases, the average (absolute) cosine similarity of the 10 most similar (or dissimilar)

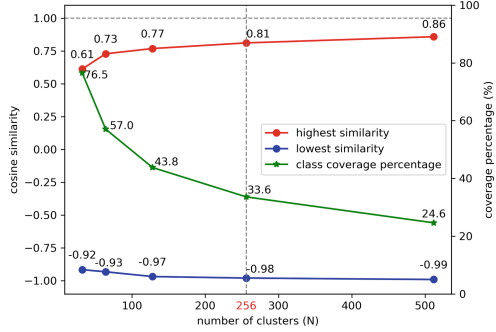


Fig. 7. Effect of the number of clusters.

analogy pairs increases smoothly, which is in line with intuitions and indicates better consistency between analogies. Empirically, we observe a trade-off between generality and interpretability of the analogies, which can be controlled by  $N$ . Specifically, a smaller  $N$  leads to more general analogies, but at a cost of their interpretability to humans. For example, in our preliminary experiments, when  $N$  is set to 32 and 64, the clusters include image pairs from more diverse classes in our visualization, showing better generality across classes. However, they are more difficult to interpret. As  $N$  continues to grow, the learned clusters become increasingly interpretable. However, they are also more class-specific, which may compromise generality. To show this trade-off, we randomly sample 1,048,576 ( $2^{20}$ ) image pairs, assign them to the nearest clusters, and compute the percentage of classes that the learned clusters cover. We report the average class coverage of the learned clusters in Fig. 7. It shows that the coverage percentage decreases as  $N$  increases, which is in line with our observation. We believe that an optimal  $N$  is also dependent on the dataset to be explored. We find that 128 and 256 clusters achieve a reasonable trade-off on ImagNet. Additionally, there is no significant difference in convergence time.

## 5 Limitations and Future Work

Although we show that CLIP space possesses the analogical ability as found in other well-established models [27, 29], and propose an effective way to extract such image analogies in the latent space, there are some limitations.

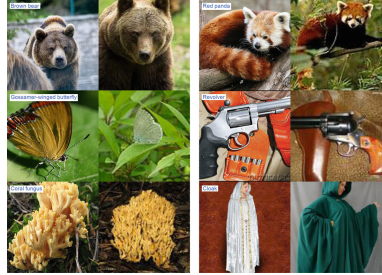
**Choice of Image Datasets.** We look into some clusters that do not present a clear interpretable analogy, and find that these clusters usually contain image pairs with almost identical semantics. One example is provided in Fig. 8. Such clusters can be interpreted as learning an *identity* analogy, which transitions

the source image to a target image that is identical in semantics. We argue that this is partly due to the fact that ImageNet contains many images that are visually similar. Therefore, some clusters may learn a shortcut by capturing analogies between these similar images when clustering is performed without any constraint.

This could be solved by preprocessing the dataset and filtering out visually similar images within each class. Additionally, exploring image analogies in the CLIP space based on one annotated dataset means that the learned analogies are limited to this dataset. If an analogy cannot be derived from the dataset, it will not be discovered with our proposed strategy. In the future, we will employ our clustering-based analogy discovering approach to other large-scale datasets. It is also possible to employ our method to explore general analogies on web-scale image-text data without class annotations. To ensure that the analogies are generalisable across visual concepts, *i.e.*, they can be applied to an image to change the semantics without altering the visual concept present, it is necessary to first identify concepts in the data, and categorise the images based on these concepts. A possible solution could be extracting concepts from the corresponding texts by applying named entity recognition. An advantage of extracting concepts from web-scale data over pre-defined classes of annotated datasets is that more diverse analogies could be derived from unlimited visual concepts. We leave this exploration as part of future work.

**Assumptions.** We propose to explore image analogies based on the parallelogram assumption, which has been shown to be able to explain some analogies in word embedding models [12]. Albeit intuitive, the parallelogram model has recently been shown to be better at capturing some word relations than others [8, 21, 30]. We believe that more advanced analogy models can be helpful in explaining and learning analogies in the vector space.

**Evaluation of Analogies.** Just as studies in cognitive science are largely dependent on human participants, in this work, we rely on human evaluation to interpret the analogies captured by the clusters. Although in principle, humans should provide higher-quality evaluation than automatic metrics, it can be labour-intensive. Furthermore, interpretability of the learned clusters to humans does not necessarily relate to how the representations are interpreted in the CLIP space. One potential direction in the future is to assess image analogies in the CLIP space based on text that explicitly describes these semantic changes. We note that a recently proposed task, termed *set difference captioning* [11], can be combined seamlessly with our analogy discovery strategy. Specifically, Dunlap *et al.* [11] propose to leverage multiple vision-language foundation mod-



**Fig. 8.** A cluster with no clear interpretable analogy, visualized by six image pairs sampled from this cluster. All images are cropped to square sizes.



els compositionally, including BLIP-2 [25], GPT-4 [1], and CLIP [31], to generate descriptions that best separate two sets of images. As part of future work, we are interested in combining our work with this line of research to automatically interpret and evaluate the learned clusters.

**Future Research Directions.** This work initiates an exploration into the analogical reasoning abilities of the CLIP latent space. We believe that this work is beneficial to a variety of research directions, which we summarise below: (1) Our work can be employed to explore the latent space of other foundation models and further advance understanding of the properties of these widely-deployed models, which is currently understudied. (2) In this study, we find analogies that represent high-level semantic changes, *e.g.*, *removal of written words* and *condition shift* as shown in Fig. 5d and Fig. 5f, respectively. Such high-level analogies are beneficial to various research questions. For example, CLIP has been shown to be sensitive to typographical attacks due to the entanglement of visual and written concepts [4, 16, 26]. However, the analogy of *removal of written words* shows that there exists a hyperplane in the latent space determined by this analogy vector that may be helpful in separating written and visual concepts. Likewise, analogies of *condition shift* can be employed to manipulate given images to exhibit specific conditions. Therefore, one research direction in the future is to employ these analogies to related tasks. (3) Our study can be employed to enrich the semantics of an image. For example, given an image at test time, it is possible to employ the learned analogies to transform the semantics of the image without altering the object. This is especially beneficial to scenarios where training data are scarce, *e.g.*, in a few-shot learning setting.

## 6 Conclusion

In this work, we analyse the analogical reasoning ability of the CLIP space, which has been found in word embedding models and the feature space of ConvNet feature extractors. We show that CLIP indeed possesses the property of making analogies with respect to the semantics of images, by using simple linear vector arithmetic operations in the CLIP space. An effective clustering based strategy is proposed to discover these general image analogies irrespective of their classes. We show that most of the analogies captured by the clusters are interpretable to humans through a visualisation experiment. To our best knowledge, no prior research efforts have been devoted to exploring the ability of analogical reasoning in the CLIP space. We hope that this work inspires future research in the understanding of CLIP representations and the exploration of human-like reasoning abilities in the CLIP space.

## References

1. Achiam, J., et al.: GPT-4 technical report. arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774) (2023)
2. Allen, C., Hospedales, T.: Analogies explained: towards understanding word embeddings. In: International Conference on Machine Learning, pp. 223–231. PMLR (2019)

3. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: A latent variable model approach to PMI-based word embeddings. *Trans. Assoc. Comput. Linguist.* **4**, 385–399 (2016)
4. Azuma, H., Matsui, Y.: Defense-prefix for preventing typographic attacks on clip. arXiv preprint [arXiv:2304.04512](https://arxiv.org/abs/2304.04512) (2023)
5. Bar, A., Gandselman, Y., Darrell, T., Globerson, A., Efros, A.: Visual prompting via image inpainting. *Adv. Neural. Inf. Process. Syst.* **35**, 25005–25017 (2022)
6. Bitton, Y., Yosef, R., Strugo, E., Shahaf, D., Schwartz, R., Stanovsky, G.: VASR: visual analogies of situation recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 241–249 (2023)
7. Brown, T., et al.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
8. Chen, D., Peterson, J.C., Griffiths, T.L.: Evaluating vector-space models of analogy. *CoRR* [arXiv:abs/1705.04416](https://arxiv.org/abs/1705.04416) (2017)
9. 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 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
10. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021)
11. Dunlap, L., et al.: Describing differences in image sets with natural language. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24199–24208 (2024)
12. Ethayarajh, K., Duvenaud, D., Hirst, G.: Towards understanding linear word analogies. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3253–3262. Association for Computational Linguistics, Florence, Italy (2019)
13. Gandselman, Y., Efros, A.A., Steinhart, J.: Interpreting CLIP’s image representation via text-based decomposition. In: *The Twelfth International Conference on Learning Representations* (2024). <https://openreview.net/forum?id=5Ca9sSzuDp>
14. Gentner, D.: Structure-mapping: a theoretical framework for analogy. *Cogn. Sci.* **7**(2), 155–170 (1983)
15. Gittens, A., Achlioptas, D., Mahoney, M.W.: Skip-Gram – Zipf + uniform = vector additivity. In: Barzilay, R., Kan, M.Y. (eds.) *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 69–76. Association for Computational Linguistics, Vancouver, Canada (2017)
16. Goh, G., et al.: Multimodal neurons in artificial neural networks. *Distill* **6**(3), e30 (2021)
17. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027 (2017)
18. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 327–340. SIGGRAPH 2001, Association for Computing Machinery, New York, NY, USA (2001)
19. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.* **23**(1), 2249–2281 (2022)



20. Holyoak, K.J.: Analogy and Relational Reasoning. In: *The Oxford Handbook of Thinking and Reasoning*, pp. 234–259 (2012)
21. Hummel, J.E., Doumas, L.A.A.: *Analogy and Similarity*, p. 451–473. Cambridge Handbooks in Psychology, Cambridge University Press, 2nd edn. (2023). <https://doi.org/10.1017/9781108755610.018>
22. Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., Heming, J.: K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data. *Inf. Sci.* **622**, 178–210 (2023)
23. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: *International Conference on Machine Learning*, pp. 4904–4916. PMLR (2021)
24. Lemesle, Y., Sawayama, M., Valle-Perez, G., Adolphe, M., Sauzéon, H., Oudeyer, P.Y.: Language-biased image classification: evaluation based on semantic representations. In: *International Conference on Learning Representations (ICLR)* (2022)
25. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: *International conference on machine learning*, pp. 19730–19742. PMLR (2023)
26. Materzyńska, J., Torralba, A., Bau, D.: Disentangling visual and written concepts in clip. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16410–16419 (2022)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *ICLR* (2013)
28. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
29. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014)
30. Peterson, J.C., Chen, D., Griffiths, T.L.: Parallelograms revisited: exploring the limitations of vector space models for simple analogies. *Cognition* **205**, 104440 (2020)
31. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR (2021)
32. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: *International Conference on Learning Representations (ICLR)* (2016)
33. Reed, S.E., Zhang, Y., Zhang, Y., Lee, H.: Deep visual analogy-making. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc. (2015)
34. Richland, L.E., Simms, N.: Analogy, higher order thinking, and education. *WIREs Cognit. Sci.* **6**(2), 177–192 (2015)
35. Rumelhart, D.E., Abrahamson, A.A.: A model for analogical reasoning. *Cogn. Psychol.* **5**(1), 1–28 (1973)
36. Sculley, D.: Web-scale k-means clustering. In: *Proceedings of the 19th International Conference on World Wide Web*, pp. 1177–1178 (2010)
37. Šubrťová, A., Lukáč, M., Čech, J., Futschik, D., Shechtman, E., Sỳkora, D.: Diffusion image analogies. In: *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–10 (2023)

38. Ushio, A., Espinosa Anke, L., Schockaert, S., Camacho-Collados, J.: BERT is to NLP what AlexNet is to CV: can pre-trained language models identify analogies? In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3609–3624. Association for Computational Linguistics, Online (2021)



# A Framework for Mining Collectively-Behaving Bots in MMORPGs

Hyunsoo Kim<sup>(✉)</sup>, Jun Hee Kim, Jaeman Son, Jihoon Song, and Eunjo Lee<sup>(✉)</sup>

NCSOFT, Seongnam-si, Republic of Korea

{aitch25,junheekim,jaemanson,songjh7919,gimmesilver}@ncsoft.com

**Abstract.** In MMORPGs (Massively Multiplayer Online Role-Playing Games), abnormal players (bots) using unauthorized automated programs to carry out pre-defined behaviors systematically and repeatedly are commonly observed. Bots usually engage in these activities to gain in-game money, which they eventually trade for real money outside the game. Such abusive activities negatively impact the in-game experiences of legitimate users since bots monopolize specific hunting areas and obtain valuable items. Thus, detecting abnormal players is a significant task for game companies. Motivated by the fact that bots tend to behave collectively with similar in-game trajectories due to the auto-programs, we developed BotTRep, a framework that comprises trajectory representation learning followed by clustering using a completely unlabeled in-game trajectory dataset. Our model aims to learn representations for in-game trajectory sequences so that players with contextually similar trajectories have closer embeddings. Then, by applying DBSCAN to these representations and visualizing the corresponding moving patterns, our framework ultimately assists game masters in identifying and banning bots.

**Keywords:** Gaming bot detection · Trajectory representation model

## 1 Introduction

In MMORPGs (Massively Multiplayer Online Role-Playing Games), player activities naturally generate diverse patterns, similar to those in the real world. They can undertake various tasks individually or with others. Furthermore, there are groups within the game that carry out activities with malicious intent, as in the real world. The collective behaviors of bots, which exhibit abnormal gaming efficiency due to auto-programs, negatively impact the in-game experiences of regular players. Bots not only monopolize many aspects of the game but also participate in real money trading, which disrupts the in-game economy [8, 12].

In this study, we introduce a framework for mining collectively-behaving bots, one of the most prevalent forms of abuse in MMORPGs inspired by

---

J. H. Kim and J. Son—Equal contribution.

moving-together patterns in the real world [2, 3, 7, 13, 21, 23, 24, 31]. However, the collectively-behaving groups we aim to identify differ from real-world patterns due to the unique behaviors of bots, such as automatic and sporadic actions needed for purchasing potions, strategic hunting, and returning from dying. Additionally, teleportation in MMORPGs complicates the detection of these patterns, skewing results that traditional real-world methods might yield. Consequently, our defined collectively-behaving clusters comprise groups of players who not only engage in synchronized activities to optimize farming efficiency but also display suspicious sporadic behaviors driven by situational demands.

Despite the crucial need for fast and accurate bot detection from a service perspective, identifying these bots based on deep learning models is challenging for several reasons: 1) Real-time labeling of the various trajectories observed in new forms daily is difficult. 2) To establish sufficient evidence, observing whether group movements occur for at least an entire day is necessary, resulting in long sequences. 3) Furthermore, to ensure stable service operation, we must begin monitoring at 9 AM the day after an update to reflect any newly added regions in the game. This means that the training time must be within 9 h. 4) False positive detections can bring a loss of trust from users and also cause legal issues (e.g. lawsuit after falsely banning a benign user), and hence an effective methodology for a comprehensive understanding of many users is required. Therefore, we propose a framework to effectively address these industrial challenges, with our contributions as follows:

- This framework mines collectively-behaving bots even without labels, proposing a method for trajectory representation learning and DBSCAN [5].
- The model is designed for efficiency, allowing it to train on long datasets covering an entire day in a shorter time compared to traditional models.
- An effective visualization methodology is proposed to quickly double-check if the detected group activity patterns are genuinely collective, contributing to more precise operations.

To demonstrate the performance and design validity of our model, we primarily use actual gameplay data from Lineage W<sup>1</sup>, which is an MMORPG released by NCSOFT in November 2021 and is ranked 1 in “Top Grossing Games Worldwide for H1 2022” (Google Play Revenue)<sup>2</sup>.

We refer to the proposed model as “**BotTRep**,” which stands for a **T**rajectory **R**epresentation model designed to mine **B**ots in the game world.

## 2 Background

### 2.1 Trajectory Data Mining for Real World Tasks

**Related Works.** There are various research fields and applications in trajectory data mining [3, 4, 6, 13, 21, 23–25, 27–29, 32, 33]. Even though our research

<sup>1</sup> <https://lineagew.plaync.com>.

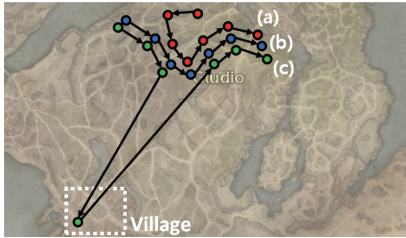
<sup>2</sup> <https://sensortower.com/blog/app-revenue-and-downloads-1h-2022>.

aligns closely with studies such as [13, 21, 24], we had to design a new mechanism because our research objectives differ from those of trajectory mining in two key aspects. Firstly, the collectively-behaving bots we aim to detect are not merely groups with similar movement trajectories. While there may be subsets within the collectively-behaving bots that have generally similar trajectories, the bots we need to identify exhibit sporadic behaviors, such as some players replenishing potions in the village or returning to the hunting ground after being killed. Bots have diverse patterns of collective behaviors. Consequently, methods that model representations based on Euclidean space, assuming that two trajectories are similar if they are close in time and space, are fundamentally inappropriate for MMORPGs. Moreover, the coordinate systems in MMORPGs are based on a local coordinate system, making spatial features in Euclidean space incompatible. These are elaborated in the following paragraph. Secondly, we needed to focus not only on how to extract appropriate trajectory representations but also on how to efficiently train on lengthy sequences. However, studies addressing real-world problems [13, 21, 24] primarily focus on solving issues related to shorter sequences, without considering the challenges posed by longer sequences.

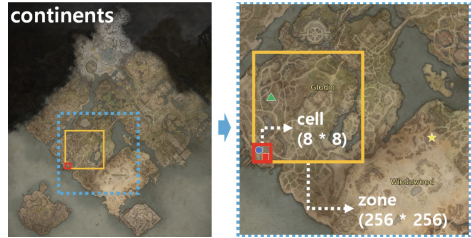
**Data Compatibility Issues in Two Worlds.** Several issues arise when applying representation learning proposed for real-world data to our task data. The first problem is the existence of teleportation in MMORPGs. Teleportation, a technique that allows for instant movement between two distant spaces, completely overturns our conventional understanding of “distance”. For instance, players use teleportation to travel from the village to a hunting ground and briefly return to the village to purchase potions during hunting. Thus, while the distance between the village and the hunting ground may not be short, from the perspective of actual behavior in context, they are relatively close. Figure 1 provides an example of a common trajectory challenge encountered in Lineage W. In this scenario, entities (b) and (c) are collectively-behaving bots, with entity (c) having died during hunting, revived in the village, and then automatically returned to the hunting grounds. In contrast, entity (a) represents a player with no affiliation to (b) and (c), whose path coincidentally overlaps with that of (b) for a portion of their movement. In this context, the question arises regarding which entity, (a) or (c), should have embeddings more closely aligned with (b). Models based on spatial features would likely find the embeddings of (a) and (b) more similar due to the physical proximity of their trajectories. However, considering the behavioral context of (b) and (c), their embeddings should be closer. Considering this, we implemented our model by leveraging the contextual relationship between regions instead of Euclidean distance. Thus, in this study, two different spaces frequently visited by the same player are considered “close” contextually.

Secondly, the coordinate systems of the two worlds are different. For example, consider two different continents on Earth. The coordinates between the first and the second continents share a coordinate system similar to the concept of a global coordinate system. However, the case is slightly different in

MMORPGs. In MMORPGs, for the convenience of game design, some spaces are implemented to have a local coordinate system. For instance, in Lineage W, many instance dungeons share the same coordinate range. That is, although each instance dungeon is an independent space, their coordinate ranges overlap, and players in different instance dungeons are recorded as they were in the same space. This means that it is impossible to use the coordinate features defined on the assumption of spatial proximity as suggested in [13, 21, 24] directly.



**Fig. 1.** Spatially, entities (a) and (b) are in close proximity; however, if the overall context between (b) and (c) is similar, then the embeddings of (b) and (c) should be closer.



**Fig. 2.** This figure represents the outcome of binning applied to coordinate values in Lineage W. We bin coordinate values into zones and cells of different sizes and embed them.

## 2.2 Bot Detection and Trajectory Mining for MMORPGs

**Related Works.** The mentioned works utilize various features for classifying abnormal players, such as logs from player’s status (portrait), events, quests, tradings, mouse clicks, and trajectory [1, 12, 15, 16, 19, 20, 30]. Amongst the works, this study aimed at trajectory mining for bot detection in MMORPGs [1, 15, 19, 30]. These works focused on detecting bots using their in-game trajectories with various approaches. For instance, [1, 15] designed models to classify the bots from benign players inspired by repetitive and regular observations generated by bots with various in-game features, such as lingering, smoothness, detour route, and turn angle. On the other hand, [19, 30] utilized deep-learning-based approaches to classify the players with suspicious trajectory patterns.

Notably, compared to the other related works, our framework puts emphasis on providing explainable materials for game masters. For example, authors in [1, 15, 19, 30] focused only on constructing accurate models to classify bots and benign players using their trajectories and other features. Consequently, their features were preprocessed into an unexplainable form, which is suitable for deep learning models but challenging for human beings to understand. Particularly, the authors in [19, 30] dynamically set the size of areas corresponding to each trajectory token based on the frequency of visits to each region and utilized preprocessed time information with diverse event types, such as finger touches or

mouse clicks. These features showed respectable performances in downstream tasks but were complicated for model explanation.

**The Necessity of Explainability in Industrial Applications.** The current trend focuses on automation using deep learning, and some companies aim to exclusively use deep learning models for automatically sanctioning abnormal users. Most studies related to gaming bots have concentrated on how to detect the bots with high accuracy. Their research is significant; however, in practical industries, there is a persistent question not just about whether a specific player group is bots, but why they are bots. When there is clear evidence, game masters can go ahead and ban the detected users with lower legal risk. Therefore, in the context of bot detection, the interpretability of model results and the ability to provide evidence are crucial. Especially in real world scenarios, if game masters mistakenly ban legitimate users, they can face legal repercussions, and the company may set unfavorable precedents. Moreover, even if game masters have properly sanctioned bots, failing to provide evidence for the reasons they were classified as bots when the bots' owners file a lawsuit also sets a bad precedent. For this reason, we propose a framework that does not fully automate the process but instead aids game masters in more efficiently detecting abnormal players.

## 3 Proposed Approach

### 3.1 Preliminary

**Defining Areas.** In this section, we introduce three geographical terms to distinguish areas in the game world: continent, zone, and cell. A continent is the largest area category, classified by whether players can move on foot or need to use a portal or teleport. For example, islands and instance dungeons are treated as separate continents. We define zones by dividing continents into multiple areas, each sized at 256 by 256 coordinates, as shown in Fig. 2. Similarly, zones contain several cells, each sized at 8 by 8 coordinates. In Lineage W, players can move about 256 coordinates per minute. The main continent measures 2048 by 2048 coordinates, and there are over 100 continents, including islands and instance dungeons, each sized between 256 by 256 and 512 by 512. When logging location coordinates in the game, the unique continent ID where the player was located at each timestamp and the detailed coordinates within that continent are recorded.

Our model is trained by zone and cell tokens, which provide spatial information. Zones offer abstract representations for larger areas, while cells provide specific details for smaller areas. Training the model with only zone tokens reduces the out-of-vocabulary issue but hampers its ability to distinguish between different trajectories. Conversely, training with only cell tokens allows discrimination between trajectories but leads to the out-of-vocabulary problem and unstable convergence in model training.

We determined the appropriate criteria empirically during model design. We recommend setting the width and height of a zone to be half the size of an instance dungeon to prevent tokens from being overly abstracted. The width and height of a cell should be approximately the range of a ranged character, such as a mage or archer. This ensures that the movement of ranged characters, as well as groups of both ranged and melee characters, can be detected more precisely.

**The Definition of Collectively-Behaving Bots.** As outlined earlier, our goal is to detect collectively behaving bots as accurately and extensively as possible. The term “collectively-behaving bots” refers to groups of 4 or more players exhibiting evidence of group activities throughout their session. Our proposed framework is designed to identify such behavior clusters.

**Contrastive Model.** We propose a model for detecting collectively-behaving bots using a contrastive approach. The model learns to make the representations of similar trajectory inputs closer together while pushing the representation vectors of dissimilar trajectory inputs further apart. The reasons for choosing a contrastive model are clear. First, contrastive models yield more robust trajectory representations [26]. Second, the task requires faster training times. Existing models for real-world problems [13, 21, 24] typically use autoencoder structures, which lack a direct procedure for distinguishing similar and dissimilar sequence pairs. Additionally, autoencoders are heavy and slow due to their encoder-decoder structure. While complex autoencoder structures are suitable for tasks that require understanding precise relationships between tokens, our task prioritizes extracting appropriate representations of trajectory sequences over token relationships. Thus, we propose a lightweight contrastive model that ensures faster convergence and superior performance specifically for this task.

### 3.2 Data Preparation

**Training Dataset.** The game logs we used in this work consist of coordinates and timestamp logs sampled at one-minute intervals. This means that if a user played the game for an entire day, we would sample 1,440 logs—one for each minute of the day.<sup>3</sup> As mentioned earlier, the game logs we aim to train on are significantly lengthy. Consequently, instead of feeding the entire log sequence into the model for training, we have preprocessed the structure of the input data to ensure the model can effectively learn the relationships between contextually close cells appearing in each sequence. To achieve this, we extract a data point in the training dataset based on the following rules:

1. Collect data for all locations where players have visited on a daily basis.
2. Utilize the collected coordinates logs to generate tokens for zones and cells.

---

<sup>3</sup> 24 hours a day is 1,440 minutes.



3. Construct a sequence using the generated zone and cell tokens, ensuring that neighboring cells within the sequence are not identical, to include various local information within a sequence.
4. Split the sequence into multiple data points, each with a length of 32.
5. Reorganize the preprocessed data with a length of 32 into triplet formats in two modes: 1) odd-even split mode, which uses odd and even indexes, and 2) half split mode, which uses the first half and second half indexes. In each mode, the anchor ( $\mathcal{A}$ ), positive ( $\mathcal{P}$ ), and negative samples ( $\mathcal{N}$ ) will have a length of 16 each.
6. Finally, masking is applied to the preprocessed anchor sequences from the previous step. The masking occurs with a probability of  $r$  (where  $r \in \{0.2, 0.3\}$ ) for the sequence tokens.

Specifically, we preprocessed the training sequences to a length of 16 to ensure the model effectively learns the differences between token sequences in each positive and negative sample. When constructing anchor, positive, and negative samples based on two split modes, a longer input sequence would contain too much regional information, causing the token types in positive and negative samples to become similar. Additionally, the similarity of the trajectories between the anchor and positive samples would decrease, especially in the half-split mode. Consequently, the model would struggle to clearly learn the differences between positive and negative samples.

To express the process mathematically, we first define ( $L^{p_1}$ ) in equation (1) as the raw coordinate location sequence of a specific player  $p_1$ , where  $p_1 \in \mathbf{P}$ , and  $\mathbf{P}$  is the entire set of players. Here,  $\{p_1, \dots, p_N\} = \mathbf{P}$ , and  $N$  is the number of players. An element  $l_i$  (where  $l_i \in L^{p_1}$ ) is in the form of  $(x, y)$  coordinate pair with continent ID ( $c$ ):  $(l_i^{(x)}, l_i^{(y)}, l_i^{(c)})$ . The continent ID addresses the design issue of the local coordinate system, where two players located in different spaces could be recorded as being at the same coordinates.

$$L^{p_1} = \left\{ (l_1^{(x)}, l_1^{(y)}, l_1^{(c)}), (l_2^{(x)}, l_2^{(y)}, l_2^{(c)}), \dots, (l_{|L^{p_1}|}^{(x)}, l_{|L^{p_1}|}^{(y)}, l_{|L^{p_1}|}^{(c)}) \right\} = \{l_1, l_2, \dots, l_{|L^{p_1}|}\} \quad (1)$$

Now, we generate a sequence for representation learning by selecting some elements in  $L^{p_1}$  corresponding to the conditions in equation (2). Equation (2) is included to ensure that diverse cell tokens are incorporated into a single sequence. Then, we define  $S^{p_1}$  in (3) as entire lengths of preprocessed binned sequences of  $p_1$ . The elements of  $L^{p_1}$  are binned into bins named zone and cell by applying two functions:  $\mathbf{z}(l'_i) = (\lfloor l'_i{}^{(x)}/256 \rfloor, \lfloor l'_i{}^{(y)}/256 \rfloor, l'_i{}^{(c)})$ , and  $\mathbf{c}(l'_i) = (\lfloor l'_i{}^{(x)}/8 \rfloor, \lfloor l'_i{}^{(y)}/8 \rfloor, l'_i{}^{(c)})$ .

$$L'^{p_1} = \{l_i \mid \mathbf{c}(l_{(i-1)}) \neq \mathbf{c}(l_i) \text{ or } i = 1, \text{ for } i \in \{1, 2, \dots, |L^{p_1}|\}\} \quad (2)$$

$$S^{p_1} = \left\{ (\mathbf{z}(l'_i), \mathbf{c}(l'_i)) \mid l'_i \in L'^{p_1}, i = 1, 2, \dots, |L'^{p_1}| \right\} \quad (3)$$

where  $L^{p_1}$ ,  $L'^{p_1}$ , and  $S^{p_1}$  are ordered set.

Here, we rewrite elements of  $S^{p_1}$  as  $S^{p_1} = \{(\mathbf{z}(l'_1), \mathbf{c}(l'_1)), (\mathbf{z}(l'_2), \mathbf{c}(l'_2)), \dots, (\mathbf{z}(l'_{|L'^{p_1}|}), \mathbf{c}(l'_{|L'^{p_1}|}))\} = (s_1, \dots, s_{|L'^{p_1}|})$  for readability. Afterward, we split the preprocessed  $S^{p_1}$  into  $j$  subsequences, each with a length of 32 in equation (4).

$$S_j^{p_1} = (s_{(j-1) \cdot 32 + 1}, s_{(j-1) \cdot 32 + 2}, \dots, s_{j \cdot 32}) \text{ where } j \text{ is an integer, } 1 \leq j \leq \left\lfloor \frac{|L'^{p_1}|}{32} \right\rfloor \quad (4)$$

Through this process, we preprocessed the entire daily coordinate sequence of a specific user into sequences of length 32 ( $S_j^{p_1}$ ). This process is then repeatedly applied to all users  $\{p_1, \dots, p_N\}$ . The result can be represented as  $D^{prep} = \{S_1^{p_1}, \dots, S_{\lfloor |L'^{p_1}|/32 \rfloor}^{p_1}, \dots, S_1^{p_N}, \dots, S_{\lfloor |L'^{p_N}|/32 \rfloor}^{p_N}\} = \{d_1^{prep}, \dots, d_M^{prep}\}$  where  $M$  is the number of preprocessed data point. Afterward, we reorganize the elements  $d_k^{prep}$  (where  $d_k^{prep} \in D^{prep}$ ,  $1 \leq k \leq M$ ) into triplet formats ( $\mathcal{A}$ ,  $\mathcal{P}$ , and  $\mathcal{N}$ ) in two modes as follows:

### Odd-even split

### Half split

$$\begin{aligned} \mathcal{A}_k^{(o)} &= \{s_i \in msk_r^\delta(d_k^{prep}) \mid i = 1, 3, \dots, 31\} & \mathcal{A}_k^{(h)} &= \{s_i \in msk_r^\delta(d_k^{prep}) \mid i = 1, \dots, 16\} \\ \mathcal{P}_k^{(o)} &= \{s_i \in d_k^{prep} \mid i = 2, 4, \dots, 32\} & & \\ \mathcal{N}_k^{(o)} &= \mathcal{P}_{k'}^{(o)} \text{ where } k' \sim U(1, M) & \mathcal{P}_k^{(h)} &= \{s_i \in d_k^{prep} \mid i = 17, \dots, 32\} \\ & \quad k \neq k' & \mathcal{N}_k^{(h)} &= \mathcal{P}_{k'}^{(h)} \text{ where } k' \sim U(1, M) \\ & & & \quad k \neq k' \end{aligned} \quad (5)$$

where  $U(1, M)$  is a uniform distribution. That is, when constructing the negative sample for index  $k$ , we composed the data by assigning the positive sample from a different sample (where  $k \neq k'$ ) out of the total  $M$  data points. Next,  $msk_r^\delta(\cdot)$  is a function that applies masking to input tokens with a probability of  $r$ , where  $r \in \{0.2, 0.3\}$ . The information about which token indices have been masked is recorded in a set  $\delta$ .

We repeat this process for entire data points ( $M$ ) to create the training dataset

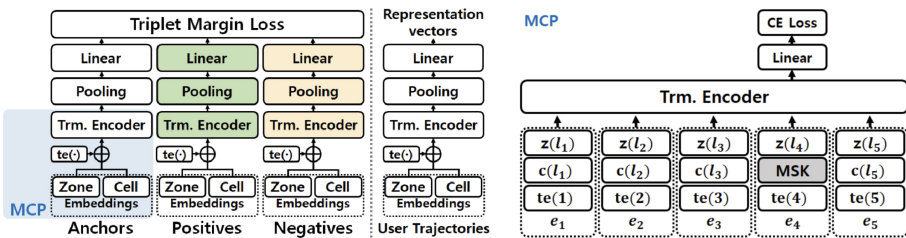
$$(D^{train}). \text{ That is, } D^{train} = \{(\mathcal{A}_1^{(o)}, \mathcal{P}_1^{(o)}, \mathcal{N}_1^{(o)}), \dots, (\mathcal{A}_{\lfloor M/2 \rfloor}^{(o)}, \mathcal{P}_{\lfloor M/2 \rfloor}^{(o)}, \mathcal{N}_{\lfloor M/2 \rfloor}^{(o)}), \dots, (\mathcal{A}_{\lfloor M/2 \rfloor + 1}^{(h)}, \mathcal{P}_{\lfloor M/2 \rfloor + 1}^{(h)}, \mathcal{N}_{\lfloor M/2 \rfloor + 1}^{(h)}), \dots, (\mathcal{A}_M^{(h)}, \mathcal{P}_M^{(h)}, \mathcal{N}_M^{(h)})\} = \{d_1^{train}, \dots, d_M^{train}\}.$$

**Dataset for Downstream Tasks.** Preprocessing the dataset for downstream tasks is much simpler compared to generating the training data. For downstream tasks, we utilize the raw trajectory sequence  $L^{p_1}$  right after applying binning functions:  $\mathcal{T}^{p_1} = \{(\mathbf{z}(l_i), \mathbf{c}(l_i)) \mid l_i \in L^{p_1}, i = 1, \dots, |L^{p_1}|\}$ . This process is repeated for all  $N$  players, and we notate this dataset as  $D^{traj} = \{\mathcal{T}^{p_1}, \dots, \mathcal{T}^{p_N}\}$ .

### 3.3 Task-Specific Representation Model

BotTRep utilizes a contrastive structure based on the Transformer architecture [11,22]. It is trained jointly using the triplet margin loss [17] and cross-entropy loss for two tasks: contrastive learning and masked cell prediction, as shown in Fig. 3. The following subsections provide a detailed explanation of the design strategy and the two tasks.

**Transformer-Based Contrastive Learning Task.** In this part, we begin by explaining the process of contrastive learning with a data point of zone and cell sequences, such as  $\mathcal{A}_i$  from  $(\mathcal{A}_i, \mathcal{P}_i, \mathcal{N}_i) = d_i^{train}$  where  $(d_i^{train} \in D^{train})$ , as described in Subsect. 3.2. Embedding modules,  $\mathbf{e}_z(\cdot)$  and  $\mathbf{e}_c(\cdot)$ , map each zone and cell token to  $d$ -dimensional vectors, and  $e_j \in \mathbb{R}^d$  where  $d \in \{256, 512\}$ . The embedding matrix associated with a sequence,  $\mathcal{A}_i = (s_1, \dots, s_j, \dots, s_{16})$ , is initialized as  $E_i = (e_1, \dots, e_j, \dots, e_{16})$  where  $e_j = \mathbf{e}_z(s_j) + \mathbf{e}_c(s_j) + \mathbf{te}(t + j)$  where  $\mathbf{te}(\cdot)$  is a function for timestamp encoding and  $t$  is a random index between 1 and 1424, corresponding to minute indexes in a day. The timestamp encoding has a similar concept as the positional encoding [22]. The timestamp encoding is also a mapper generated by:  $\mathbf{te}(j, 2m) = \sin(j/10000^{2m/d})$  and  $\mathbf{te}(j, 2m + 1) = \cos(j/10000^{2m/d})$ . Here, we limit the last random index to 1424 because the length of our input sequence is 16, and the input value for  $\mathbf{te}(\cdot)$  must not exceed 1440 because it indicates 1440 min a day. The reason that we input randomly generated timestamp values is to make the model train from diverse input for each epoch. We set the independent random timestamp for the anchor and negative sample, and for the positive sample, we set the dependent timestamp values from the anchor’s timestamp. For example, if we define a uniform random function as  $t \sim U(a, b)$ , and  $t_A, t_P$ , and  $t_N$  as randomly generated timestamps of the first index, we set  $t_A \sim U(1, 1424)$ ,  $t_P \sim t_A + U(-16, 16)$ , and  $t_N \sim U(1, 1424)$ , respectively. That is, the elements of  $\mathcal{A}_i$  are finally initialized as follows:  $e_j = \mathbf{e}_z(s_j) + \mathbf{e}_c(s_j) + \mathbf{te}(t_A + j)$ , and the same applies to  $\mathcal{P}_i$  and  $\mathcal{N}_i$ .



**Fig. 3.** The left side of the figure depicts our Transformer-based model for the contrastive learning task (left) and its representation extractor (right), respectively. The right side of the figure shows how the training for the MCP task is performed.

Now,  $E_i^{\mathcal{A}}$ ,  $E_i^{\mathcal{P}}$ , and  $E_i^{\mathcal{N}}$  are embedding matrices of anchor ( $\mathcal{A}$ ), positive ( $\mathcal{P}$ ), and negative sequences ( $\mathcal{N}$ ), respectively. In training, we modified the input by changing the ratio of anchor to positive samples to 2:1, allowing the model to learn from a wider variety of sequence combinations.

Then, embedding matrices ( $E_i^{\mathcal{A}}$ ,  $E_i^{\mathcal{P}}$ ,  $E_i^{\mathcal{N}}$ ) are input to the Transformer encoder layer. For convenience, all these processes are denoted as follows:  $F_i^{\mathcal{A}} = \mathbf{trm}(E_i^{\mathcal{A}})$  where  $\mathbf{trm}(\cdot)$  is a Transformer encoder block, and  $F_i^{\mathcal{A}}$  represents the token-unit output of the anchor sequence produced by the model. Here, the lengths of  $E_i^{\mathcal{A}}$  and  $F_i^{\mathcal{A}}$  are both 16. We set the dimensionality of the inner-layer ( $d_{inner}$ ) to 1024–2048, depending on the embedding dimension. The encoder block is composed of a stack of 8 identical layers in this work. Next, the pooling layer calculates average pooling from the output of the Transformer encoder block,  $F_i^{\mathcal{A}}$ , then, the linear layer receives the pooled vector and calculates the final output,  $\mathcal{R}_i^{\mathcal{A}} = \mathit{Linear}(\mathit{Pooling}(F_i^{\mathcal{A}}))$  where  $\mathit{Linear}(x) = xW^T + b$ .  $\mathcal{R}_i^{\mathcal{A}}$  is an example of the output representation of anchor; the positive and negative samples’ outputs,  $\mathcal{R}_i^{\mathcal{P}}$  and  $\mathcal{R}_i^{\mathcal{N}}$ , are calculated in the same way. The loss of our proposed model is obtained by the below function, named triplet margin loss (6). This function minimizes the distance between an anchor and a positive sample and maximizes the distance between an anchor and a negative sample.

$$\mathcal{L}_1(\mathcal{R}_i^{\mathcal{A}}, \mathcal{R}_i^{\mathcal{P}}, \mathcal{R}_i^{\mathcal{N}}) = \left[ \|f(\mathcal{R}_i^{\mathcal{A}}) - f(\mathcal{R}_i^{\mathcal{P}})\|_2^2 - \|f(\mathcal{R}_i^{\mathcal{A}}) - f(\mathcal{R}_i^{\mathcal{N}})\|_2^2 + \beta \right] \quad (6)$$

**Masked Cell Prediction (MCP) Task.** In addition to the contrastive learning task, our model incorporates the masked language model (MLM) task proposed in BERT [11] to refine the learning of cell tokens. However, we named it the “masked cell prediction” (MCP) task in our model because the task is no longer related to language models. To apply this, masking is performed on the sequence data before model training, and this masking is only applied to the anchor sequences, as shown in equation (5). In MCP, the problem involves predicting what the token was before being masked in the anchor sequence that has been masked during the preprocessing process.

Specifically, we apply the  $\mathit{Linear}(\cdot)$  function to  $F_i^{\mathcal{A}}$ , returned by  $\mathbf{trm}(\cdot)$ , for training. Since the length of the sequences inputted into our model is 16, for convenience, we denote this as  $F_i^{\mathcal{A}} = (f_1^{\mathcal{A}}, \dots, f_{16}^{\mathcal{A}})$ . In the MCP task, among these 16 extracted results,  $\mathit{Linear}(\cdot)$  is applied to the tokens that had been masked, and then the loss between the predicted results and the actual answers is calculated using Cross Entropy Loss. The process can be formalized as follows:

$$\hat{y}_j = \mathit{Linear}(f_j^{\mathcal{A}}) \quad \text{for each } f_j^{\mathcal{A}} \in F_i^{\mathcal{A}}, \text{ where } j \in \delta \quad (7)$$

where  $\delta$  is the set we recorded masked indexes, and  $\hat{y}_j$  is the predicted output for the  $j$ -th token in the sequence. The objective is to minimize the loss between the predicted output  $\hat{y}_j$  and the true label  $y_j$  for the tokens that were originally

masked. The loss is calculated using Cross Entropy Loss:

$$\mathcal{L}_2(y_j, \hat{y}_j) = - \sum_{k=1}^C y_{j,k} \log(\hat{y}_{j,k}) \quad (8)$$

where  $y_j$  is the true label for the  $j$ -th token,  $C$  is the number of classes (vocabularies), and  $k$  is the index for each class, ranging from 1 to  $C$ . The summation is performed over all tokens in the sequence that were masked. Ultimately, training is conducted by summing the losses calculated from the top two tasks and then performing backpropagation:  $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$ .

### 3.4 Extract Representation Vectors

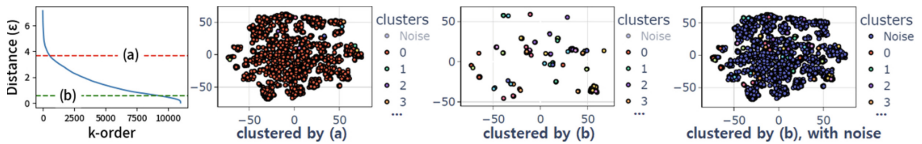
Now, we can extract representations from user trajectories using our trained model. We utilize the model that extracts user representations in Fig. 3. The entire procedure is almost the same as the model train step, but the observed timestamp values are added to each token instead of randomly generated values. We utilize timestamp encoding designed on a minute basis, as mentioned previously. Afterward, we extract the daily trajectory of each player from the games, and each data point is denoted as  $\mathcal{T}^{p_i}$ , as mentioned before. The embedding matrix associated with  $\mathcal{T}^{p_i}$  is also initialized as  $E_i^{\mathcal{T}^{p_i}} = (e_1, \dots, e_t)$  where  $e_t = \mathbf{e}_z(\mathbf{z}(l_t)) + \mathbf{e}_c(\mathbf{c}(l_t)) + \mathbf{te}(t)$  where  $t$  is timestamp index from 1 to 1440, stands for the timestamp of  $t^{th}$  location that the player has visited. Then, the embedding matrix  $E_i$  is input to the trained model, and the model extracts representations. To summarize, the simplified structure of our suggestion is  $\mathcal{T}'^{p_i} = \mathbf{model}(E_i^{\mathcal{T}^{p_i}})$  where  $\mathbf{model}(\cdot)$  is the proposed model, and the representations of the trajectories are notated in this way:  $D^{rep} = \{\mathcal{T}'^{p_1}, \mathcal{T}'^{p_2}, \dots, \mathcal{T}'^{p_N}\}$ .

### 3.5 Clustering Collectively-Behaving Groups

Once we obtain representation vectors for the trajectories, we cluster them so that players that have similar in-game trajectories and, hence, similar representations get grouped together. In particular, we use the DBSCAN [5]. An interesting property of DBSCAN is that not all data points get assigned to a cluster. That is, the algorithm classifies points that do not belong to any group as noise. Such points generally correspond to benign players because they have peculiar trajectories resulting from diverse preferences in play styles and are typically not included in specific clusters. Hence, we decided to define all the clustered groups (i.e. DBSCAN did not classify as noise) as collectively-behaving groups. However, in order to utilize DBSCAN appropriately, optimization of parameters, *min\_samples* and *eps* ( $\varepsilon$ ), should be preceded. Here, we select *min\_samples* as 4 because we target suspicious groups with more than 4 players relying on our industrial requirement. Afterward, we control  $\varepsilon$  by referring to the distances of representation vectors of 4 closest neighbors from each data point, inspired by [18]. Notably, we chose not to use a fixed  $\varepsilon$  value for DBSCAN. Instead, we

adopted the methodology from [18] that selects  $\varepsilon$  values based on the density of representation vectors. This is because the vectors we cluster are derived from the deep learning model. Even when using the same data for training, the inherent randomness in the learning process can result in representation vectors in hidden space having different scales of distance and density among specific data points. In other words, using a fixed  $\varepsilon$  value led to issues in maintaining consistent quality when detecting collectively-behaving bots based on the model’s output vectors.

Figure 4 explains how we choose the  $\varepsilon$  value in the clustering process, and the clusters that result from it. Firstly, the leftmost image shows the criterion for selecting  $\varepsilon$  as proposed in [18] (a), and the criterion we used in this work (b). [18] suggests drawing a plot of distance and then using the distance at the elbow point of the curve as the  $\varepsilon$  value. However, applying the value from (a) to our data led to the phenomenon of clustering together trajectories that are relatively dissimilar, as shown in the second plot in Fig. 4. Consequently, we searched for a new criterion (b) suitable for our data and are utilizing this value as the  $\varepsilon$  for DBSCAN. Precisely, we extract the distances between each data point and their 4 nearest data points. Afterward, we select a distance value of 0.05–0.20 quantile ( $q$ ) from the extracted distances by K-NN. For example,  $\varepsilon = \text{quantile}_{0.05}(\text{dist})$  where  $\text{quantile}_q(\cdot)$  is a quantile function that returns  $q$  quantile from input data, and  $\text{dist} = 4 - NN_{\text{dist}}(D^{\text{rep}})$  where  $4 - NN_{\text{dist}}(\cdot)$  is a  $k - NN$  algorithm where  $k = 4$  and returns distances between the 4 closest neighbors from each data point.



**Fig. 4.** The first image shows a comparison between the  $\varepsilon$  selection criterion proposed by [18] (a) and our optimized criterion (b). The third image illustrates clustering results using criterion (b), visualized with t-SNE [14] for dimension reduction, excluding noise clusters.

## 4 Experiments

### 4.1 Dataset : Lineage W

We train and evaluate our proposed model with two different datasets: 1) a preprocessed dataset for model training, and 2) a real-world gaming trajectory dataset for the downstream task. The first dataset includes 778,656 samples of preprocessed trajectories collected for 8d on July 1st-8th, 2023. When training, we performed parallel computing on 8 NVIDIA A40 GPUs. We have configured the model to be trained for at least 70 epochs and terminated based on early

stopping criteria with patience of 8 epochs. The training time was approximately 6–12 minutes per epoch, depending on the parameters. The second dataset contains 26,136 player trajectories collected for 7 d on July 9th–15st, 2023. Tables 1 and 2 present the experimental results for the second dataset.

## 4.2 Evaluation Methods

**Contextual Similarity.** In this work, we evaluate the similarity in representation between two pairs: collectively-behaving pairs and random pairs, which we call positive and negative pairs, respectively. In this step, we prepared the experimental environment by collecting daily trajectory sequences from all players in a game world and extracting their representations. Subsequently, we generated positive and negative groups based on their representations using DBSCAN. Then, we excluded noise-labeled data determined by the DBSCAN algorithm. We then selected positive and negative pairs based on the cluster labels in this way:

$$pos = \left\{ (\mathcal{T}^{p_i}, \mathcal{T}^{p_j}) \mid i = 1, \dots, N, j \in \xi(\mathcal{T}'^{p_i}) \right\} \quad (9)$$

$$neg = \left\{ (\mathcal{T}^{p_i}, \mathcal{T}^{p_j}) \mid i = 1, \dots, N, j \notin \xi(\mathcal{T}'^{p_i}) \right\} \quad (10)$$

where  $\xi(\cdot)$  is a function that returns the closest index from neighboring players within the cluster containing each input player.

After composing experimental data pairs in the above way, we measure the contextual similarity between their trajectories using a metric named time-aware Jaccard similarity. In this study, we design time-aware Jaccard similarity to check whether the pair of trajectories have similarities over time. We calculate Jaccard similarity [9, 10] by a 30-minute subset of each trajectory using 1440 range of minute indexes ( $t$ ). This metric returns the overall similarity by averaging every 30-minute subsets between two trajectories with 15-minute shifts. The similarity scores from this metric can be interpreted as a measure of how contextually close two different users were in terms of their locations at approximately the same moments. Trajectory pairs with similar travel routes exhibit high similarity values. Contextual similarity is calculated at the cell level and is defined as follows:

$$\frac{1}{|T|} \sum J_t(\mathcal{T}^{p_i}, \mathcal{T}^{p_j}) \text{ where } t \in \{1, 16, 31, \dots, 1411\} = T$$

$$\text{and } J_t(\mathcal{T}^{p_i}, \mathcal{T}^{p_j}) = \frac{|\{tr_t^{p_i}, \dots, tr_{t+29}^{p_i}\} \cap \{tr_t^{p_j}, \dots, tr_{t+29}^{p_j}\}|}{|\{tr_t^{p_i}, \dots, tr_{t+29}^{p_i}\} \cup \{tr_t^{p_j}, \dots, tr_{t+29}^{p_j}\}|} \quad (11)$$

**Access Information Homogeneity.** The another metric is designed to verify whether there is indeed the same abuser behind the clusters mined as collectively-behaving groups. In addition to user trajectory data, our dataset records access information data for each user, such as the IP and device-sharing network mentioned in [20]. This metric aims to determine whether their access information

actually belongs to the same person. In other words, if the access information of players mined as collectively-behaving groups is identical, it signifies that they are indeed real collectively-behaving bots controlled by the same user. The specific calculation method is as follows:  $\frac{1}{|C|} \sum_{c_{id} \in C} acc\_info(c_{id})$ , where  $C$  represents all clusters excluding a noise cluster, that is, the entire collectively-behaving groups we have mined.  $c_{id}$  denotes each collectively-behaving group.  $acc\_info(\cdot)$  is a function that takes a collectively-behaving group as input and returns the number of different access information points possessed by players in that cluster. For instance, if there are 4 players within a certain cluster and their access information is interconnected, 1 is returned. However, if, upon checking the access information of the 4 players, it is found that 3 of their access information points are interconnected, but one is different, then they form 2 groups, and thus 2 is returned. That is, bots controlled by the same owner have identical access information, resulting in low access information homogeneity values. In contrast, legitimate users who operate one avatar at a time have different access information, resulting in high access information homogeneity values.

### 4.3 Ablation Study

This section summarizes the ablation study results for the proposed method. We experimented by altering the model’s key parameters. Additionally, we highlight the benefits of incorporating both zone and cell tokens and the Masked Cell Prediction (MCP) technique, assessing their impact on model performance.

To validate this, we varied parameters and documented the outcomes in experiment type (a). We compared models trained on cell inputs alone versus those trained on both zone and cell inputs, and examined the effects of incorporating MCP, with results under experiment type (b). In experiment type (c), we adjusted the clustering parameter  $q$  to observe its impact on our model and DBSCAN’s performance. As  $q$  increases, contextual similarity and homogeneity of access information decrease due to less homogeneous clusters. A larger  $q$  adopts a more lenient criterion for detecting suspicious players, while a smaller  $q$  is preferred for higher precision.

**Experimental Results.** Our proposed setting for BotTRep is shown at the top of Table 1 (†). This model showed the best performance from the perspective of contextual similarity when we conducted additional experiments adjusting  $d\_model$ ,  $d\_hid$ ,  $\beta$ , and MCP ratio based on this model; the model generally exhibited higher performances in experiment type (a). In experiment type (b), we conducted experiments by removing zone embedding and MCP one at a time from †, and a slight decrease in performance was observed in both cases regarding contextual similarity. Lastly, the results from experiment type (c) showed that as clustering  $\varepsilon$  quantile ( $q$ ) increased, contextual similarity and access information all decreased. As  $q$  increases, one can observe which clusters are included in the suspicious clusters. For example, Fig. 5 on the far right shows a sample of players included in the noise cluster. Their trajectories generally do not display



a pattern; however, there are exceptions, such as case (a) in Fig. 5. If we were to set the DBSCAN’s  $\epsilon$  value higher, groups similar to (a) might be included in the collectively-behaving group. To provide additional context for the metrics, the detected bots exemplified in (<https://youtu.be/bsFXvFBVYak>) show an average time-aware Jaccard similarity of around 0.3 across all pairs and an access information homogeneity of 1.0.

**Table 1.** The results of the ablation study are shown below, with the best performance highlighted in bold. The dataset contains 26,136 data points.

Exp	Model params					clustering $\epsilon$	Detecting quant. ( $q$ )	Contextual		Acc info
	$d_{model}$	$d_{hid}$	$\beta$	zone	MCP			$pos$	$neg$	
†	256	1024	0.5	True	0.2	0.05	928	<b>0.3625</b>	0.0005	<b>1.0079</b>
(a)	256	1024	0.5	True	0.3	0.05	921	0.3596	0.0005	1.0444
	256	1024	1.0	True	0.2	0.05	976	0.3552	<b>0.0003</b>	<b>1.0079</b>
	512	2048	0.5	True	0.2	0.05	963	0.3560	<b>0.0003</b>	1.0697
(b)	256	1024	0.5	True	0.0	0.05	918	0.3208	0.0006	1.0787
	256	1024	0.5	False	0.2	0.05	917	0.3472	0.0005	1.0551
	256	1024	0.5	False	0.0	0.05	930	0.3183	0.0006	1.1307
(c)	256	1024	0.5	True	0.2	0.10	1932	0.3203	0.0006	1.0787
	256	1024	0.5	True	0.2	0.15	3031	0.2931	0.0006	1.1094
	256	1024	0.5	True	0.2	0.20	<b>4109</b>	0.2732	0.0009	1.1363

#### 4.4 Baseline Models

In this study, we deliberated on selecting the most appropriate baseline model due to the lack of precedent representation models applied to tasks similar to ours. To compare and validate the performance of our model, we prepared three types of baseline models, as described below. These models employ the encoder-decoder structure most commonly used for sequence data representation, with internal layers implemented in three variants using Bi-GRU, Bi-LSTM and Transformer blocks. We compared our model to autoencoder-based models because they are commonly used for extracting sequence representations. Previous trajectory representation models proposed for GPS data using proximity features in the real world [13, 21, 24] have also used autoencoders. For the dataset, we trained and inferred directly using the data for downstream tasks ( $D^{traj}$ ) without additional preprocessing steps, such as checking if adjacent cells are identical. This decision was made because such preprocessing significantly slowed the model’s convergence, leading to poor performance within the time constraints of our requirements.

When preparing these models, we set  $d_{model}$  to 256 to enable a comparison with our model under similar specifications. When extracting representations,

we applied mean pooling after the input data passed through the encoder block because it achieved the best performance compared to CLS and max pooling.

**Experimental Results.** Table 2 describes the outcomes of the baseline experiments. Results from these experiments revealed that the Transformer significantly outperforms Bi-GRU and Bi-LSTM. The Transformer’s results in a contextual similarity of 0.29, slightly lower than our model, while the homogeneity of access information stands at 1.79, indicating a somewhat higher figure compared to our proposal. This suggests the inclusion of benign players within the collectively-behaving groups. Transformer could not properly distinguish the two trajectories active in the same area but at different times. From the perspectives of contextual similarity and homogeneity of access information, BotTRep, with our proposed setting (†) was found to have the highest performance. This indicates that the access information of all players within a cluster is related, signifying our model has effectively detected the bot groups we aimed to identify with high accuracy. Furthermore, BotTRep completed training in about 8 h and 30 min, achieving over 70 epochs, while the other two models failed to surpass its performance even after over 24 h of training.

**Table 2.** BotTRep showed superior performance in contextual similarity and access information. The downstream dataset contains 26,136 data points.

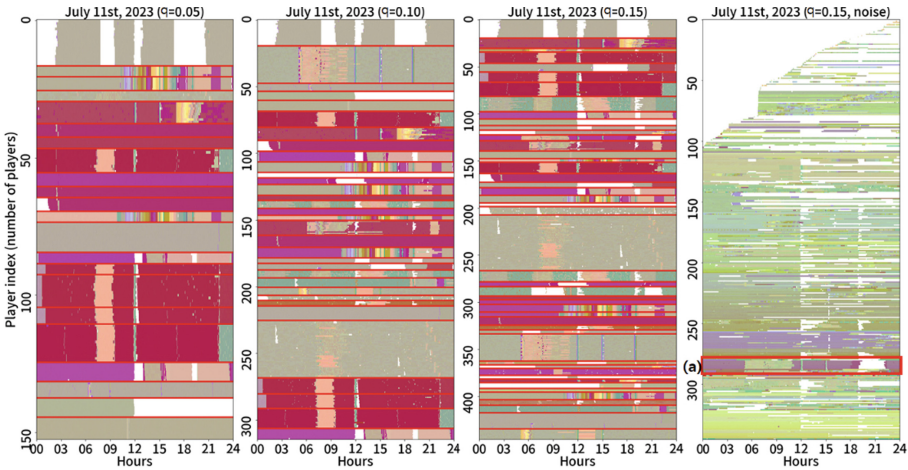
Models	Detecting count	Contextual		Acc info	Training time	Minutes (per epoch)
		pos	neg			
Bi-GRU	926	0.2157	0.0006	1.8886	24+ hours	20–21
Bi-LSTM	<b>1,064</b>	0.1716	0.0006	3.0315	24+ hours	20–21
Transformer	970	0.2921	<b>0.0002</b>	1.7855	24+ hours	25–27
BotTRep (†)	928	<b>0.3625</b>	0.0005	<b>1.0079</b>	<b>8.5 h</b>	6–7

## 4.5 Trajectory Visualization

This work is designed to surveil collectively-behaving groups, supposed to be bots, throughout the continents using their trajectory data and visualize how much their trajectories are similar to each other. The heatmap, designed to show similar colors when players exist contextually close to each other, is used to visualize whether the suspicious players appear together. The heatmap’s  $x$ -axis represents timestamps, and the  $y$ -axis represents player indices. The color, in RGB format, indicates the player’s location at a specific timestamp. That is, we construct a 3-dimensional vector that somehow represents spatial information, and then visualize it as RGB coloring. Specifically, we generate colormaps as (*continent.lv*,  $x$ ,  $y$ ), where *continent.lv* reflects average player levels in each continent. This addition represents semantic relationships between continents.

Regarding game geography, hunting grounds suitable for each level are located in an adjacency to guide players in growing their avatars with less confusion. Note that if the player was not logged in the game at a particular timestamp, we color the corresponding spot as white.

Finally, we present a visualization of player location over time in Fig. 5. The right-most plot corresponds to the points such that DBSCAN labeled as noise: i.e. those that we consider benign users. We can see that these users tend to have their own distinct trajectories, indicating that they are indeed not collectively-behaving bots. On the other hand, the three plots on the left show the results for trajectories where DBSCAN assigned a cluster. Each red horizontal line indicates the separation of clusters. Recall that players located closer to each other will exhibit similar colors in the heatmap. Clearly, we can see that players in the same cluster tend to have similar sequences of colors throughout the entire timeline ( $x$ -axis). That is, they collectively move across multiple areas or collectively log in/log out, both simultaneously and in order. This pattern is a typical characteristic observed in collectively-behaving bots that we have targeted: this arises due to multiple avatars being controlled simultaneously by automated programs connected to the same network environment. Thus, our framework that consists of trajectory embedding and clustering identifies suspicious clusters where the corresponding players move collectively.



**Fig. 5.** This image shows player locations over time based on clustering results. The  $x$ -axis represents time in minutes, the  $y$ -axis indicates individual players, and colors correspond to players' location at each time point. Red lines in the heatmap separate clusters, with similar colors denoting proximity of player locations.

## 5 Conclusion

We proposed a novel framework that uses a trajectory representation model trained jointly on contrastive learning and masked cell prediction tasks, so that similar contextual in-game movements obtain closer representations. Then, we used DBSCAN to identify collectively-behaving groups and introduced a visualization method that explains their in-game trajectories. Our framework meets the industrial needs for clear explainability and can assist game masters by providing clustered users who are suspicious to be collectively-behaving bots.

## References

1. Chen, K.-T., Liao, A., Pao, H.-K.K., Chu, H.-H.: Game bot detection based on avatar trajectory. In: Stevens, S.M., Saldamarco, S.J. (eds.) ICEC 2008. LNCS, vol. 5309, pp. 94–105. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-89222-9\\_11](https://doi.org/10.1007/978-3-540-89222-9_11)
2. Chen, L., Ng, R.: On the marriage of LP-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases-Volume 30, pp. 792–803 (2004)
3. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 491–502 (2005)
4. Chen, Z., Shen, H.T., Zhou, X.: Discovering popular routes from trajectories. In: 2011 IEEE 27th International Conference on Data Engineering, pp. 900–911. IEEE (2011)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
6. Feng, S., Cong, G., An, B., Chee, Y.M.: POI2Vec: geographical latent representation for predicting future visitors. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
7. Feng, Z., Zhu, Y.: A survey on trajectory data mining: techniques and applications. *IEEE Access* **4**, 2056–2067 (2016)
8. Huhh, J.S.: Simple economics of real-money trading in online games. SSRN 1089307 (2008)
9. Jaccard, P.: The distribution of the flora in the alpine zone. 1. *New Phytol.* **11**(2), 37–50 (1912)
10. Jadon, A., Patil, A.: A comprehensive survey of evaluation techniques for recommendation systems. arXiv preprint [arXiv:2312.16015](https://arxiv.org/abs/2312.16015) (2023)
11. Kenton, J.D.M.W.C., Toutanova, L.K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, vol. 1, p. 2 (2019)
12. Lee, E., Woo, J., Kim, H., Kim, H.K.: No silk road for online gamers! using social network analysis to unveil black markets in online games. In: Proceedings of the 2018 World Wide Web Conference, pp. 1825–1834 (2018)

13. Li, X., Zhao, K., Cong, G., Jensen, C.S., Wei, W.: Deep representation learning for trajectory similarity computation. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 617–628. IEEE (2018)
14. Van der Maaten, L., Hinton, G.: Visualizing data using T-SNE. *J. Mach. Learn. Res.* **9**(11), 2579–2605 (2008)
15. Pao, H.K., Chen, K.T., Chang, H.C.: Game bot detection via avatar trajectory analysis. *IEEE Trans. Comput. Intell. AI Games* **2**(3), 162–175 (2010)
16. Qi, X., Pu, J., Zhao, S., Wu, R., Tao, J.: A GNN-enhanced game bot detection model for MMORPGs. In: Gama, J., Li, T., Yu, Y., Chen, E., Zheng, Y., Teng, F. (eds.) *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, 16–19 May 2022, Proceedings, Part II*, vol. 13281, pp. 316–327. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-05936-0\\_25](https://doi.org/10.1007/978-3-031-05936-0_25)
17. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823 (2015)
18. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans. Database Syst. (TODS)* **42**(3), 1–21 (2017)
19. Su, Y., et al.: Trajectory-based mobile game bots detection with gaussian mixture model. In: Pimenidis, E., Angelov, P., Jayne, C., Papaleonidas, A., Aydin, M. (eds.) *Artificial Neural Networks and Machine Learning–ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, 6–9 September 2022, Proceedings, Part III*, vol. 13531, pp. 456–468. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-15934-3\\_38](https://doi.org/10.1007/978-3-031-15934-3_38)
20. Tao, J., et al.: MVAN: multi-view attention networks for real money trading detection in online games. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2536–2546 (2019)
21. Tedjopurnomo, D.A., Li, X., Bao, Z., Cong, G., Choudhury, F., Qin, A.K.: Similar trajectory search with spatio-temporal deep representation learning. *ACM Trans. Intell. Syst. Technol. (TIST)* **12**(6), 1–26 (2021)
22. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
23. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: *Proceedings 18th International Conference on Data Engineering*, pp. 673–684. IEEE (2002)
24. Wang, C., et al.: A deep spatiotemporal trajectory representation learning framework for clustering. *IEEE Trans. Intell. Transp. Syst.* **25**, 7687–7700 (2024)
25. Wang, H., Li, Z.: Region representation learning via mobility flow. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 237–246 (2017)
26. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: *International Conference on Machine Learning*, pp. 9929–9939. PMLR (2020)
27. Yan, B., Janowicz, K., Mai, G., Gao, S.: From ITDL to Place2Vec: reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In: *Proceedings of the 25th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 1–10 (2017)
28. Yao, D., Zhang, C., Huang, J., Bi, J.: SERM: a recurrent model for next location prediction in semantic trajectories. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 2411–2414 (2017)

29. Yin, Y., Liu, Z., Zhang, Y., Wang, S., Shah, R.R., Zimmermann, R.: GPS2Vec: towards generating worldwide GPS embeddings. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 416–419 (2019)
30. Zhao, S., et al.: T-Detector: a trajectory based pre-trained model for game bot detection in MMORPGs. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 992–1003. IEEE (2022)
31. Zheng, Y.: Trajectory data mining: an overview. *ACM Trans. Intell. Syst. Technol. (TIST)* **6**(3), 1–41 (2015)
32. Zheng, Y., Xie, X., Ma, W.Y., et al.: GeoLife: a collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* **33**(2), 32–39 (2010)
33. Zhou, Y., Huang, Y.: DeepMove: learning place representations through large scale movement data. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 2403–2412. IEEE (2018)



# Causal Deep Learning

M. Alex O. Vasilescu<sup>1,2</sup>(✉) 

<sup>1</sup> IPAM, University of California, Los Angeles, USA

<sup>2</sup> 10SR Vision Technologies, Los Angeles, USA

maov@cs.ucla.edu

**Abstract.** We derive a set of causal deep neural networks whose architectures are a consequence of tensor (multilinear) factor analysis, which facilitates forward and inverse causal inference. Forward causal questions are addressed with a neural network architecture composed of causal capsules and a tensor transformer. Causal capsules compute a set of invariant causal factor representations, whose interaction are governed by a tensor transformation. Inverse causal questions are addressed with a neural network that implements the multilinear projection. The architecture reverses the order of the operations of a forward neural network and estimates the causes of effects. As an alternative to aggressive bottleneck dimension reduction or regularized regression that may camouflage an inherently underdetermined inverse problem, we prescribe modeling different aspects of the mechanism of data formation with piecewise tensor models whose multilinear projections produce multiple candidate solutions. Our forward and inverse question may be addressed with shallow architectures, but for computationally scalable solutions, we derive a set of deep neural networks by taking advantage of block algebra. An interleaved kernel hierarchy implements a hierarchy of kernel tensor factor models. The resulting causal neural networks are data agnostic, but illustrated with facial images. Computational approach has been prescribed for asynchronous parallel computation.

**Keywords:** factor analysis · explanatory · latent variables · causality · tensor algebra · deep learning · generative · discriminant

## 1 Introduction

Neural networks are being employed increasingly in high-stakes application areas, such as face recognition [14, 32, 63, 64, 87], and medical diagnosis [39, 48, 67]. Developing neural networks that offer causal explanations for correct results or failures is crucial for establishing trustworthy artificial intelligence.<sup>1</sup>

---

<sup>1</sup> Causal explanations specify the causes, the mechanism, and the conditions for replicating an observed effect [23, 45, 46, 86]. Quantitatively, causality is the direct relationship between two events, A and B, where “A causes B” means “the effect of A is B”, a measurable and experimentally repeatable phenomena. Once verified with

The validity and robustness of causal explanations depend on causal model specifications in conjunction with the experimental designs used for acquiring training data [57]. Prior generative artificial intelligence research conducted by the Bengio and Hinton teams [7, 20, 50, 65] focused on unsupervised deep learning, which is not well-suited for drawing causal conclusions. They briefly addressed the connection to causal tensor factor analysis [76, 79] which models the causal mechanisms that generates data, computes invariant causal representations, and estimates both the effects of causes and the causes of effect given constraints on the solution set [72].

We derive a set of causal deep neural networks that are a consequence of causal tensor factor analysis, Figs. 1-5. Tensor factor analysis is a transparent framework for both forward [76, 78] and inverse causal inference [72, 80].<sup>2</sup>

Forward causal inference is a hypothesis-driven process, as opposed to a data-driven process, that models the mechanism of data formation and estimates the effects of interventions [35, 54, 62, 74]. This is in contrast to conventional statistics and machine learning, which model data distributions, predict one variable co-observed with another, or perform time series forecasting. Inverse causal “inference” estimates the causes of effects given an estimated forward model and constraints on the solution set [25, 72].

## 1.1 Causal Inference Versus Regression

Neural networks and tensor factorization methods may perform causal inference, or simply perform regression from which no causal conclusions are drawn. For causal inference, model specifications and experimental design for acquiring training data trump analysis [57], Fig. 2.

Causal tensor factor analysis was employed in the analysis and recognition of facial identities [74, 76], facial expressions [33], human motion signatures [17, 31, 69], and 3D sound [26]. It was employed in the transfer of facial expressions [83], and the rendering of arbitrary scenes, views and illuminations [78], etc.. Tensor factor analysis was also employed in psychometrics [8, 13, 27, 43, 68], econometrics [38, 49], chemometrics [10], and signal processing [18]. Simple tensor regression and decompositions which do not draw causal conclusions, leveraged row, column and fiber redundancies to estimate missing data [15] and to perform rank reduction [89] [11, 84] [6, 30, 37]. Recently, tensor dimensionality reduction and contractions (*i.e.*, mode- $m$  product) have been employed in machine learning to reduce neural network parameters. Network parameters are organized into “data tensors”, and dimensionally reduced [40, 41, 44, 51] [42, 53] or efficiently contracted [21].

---

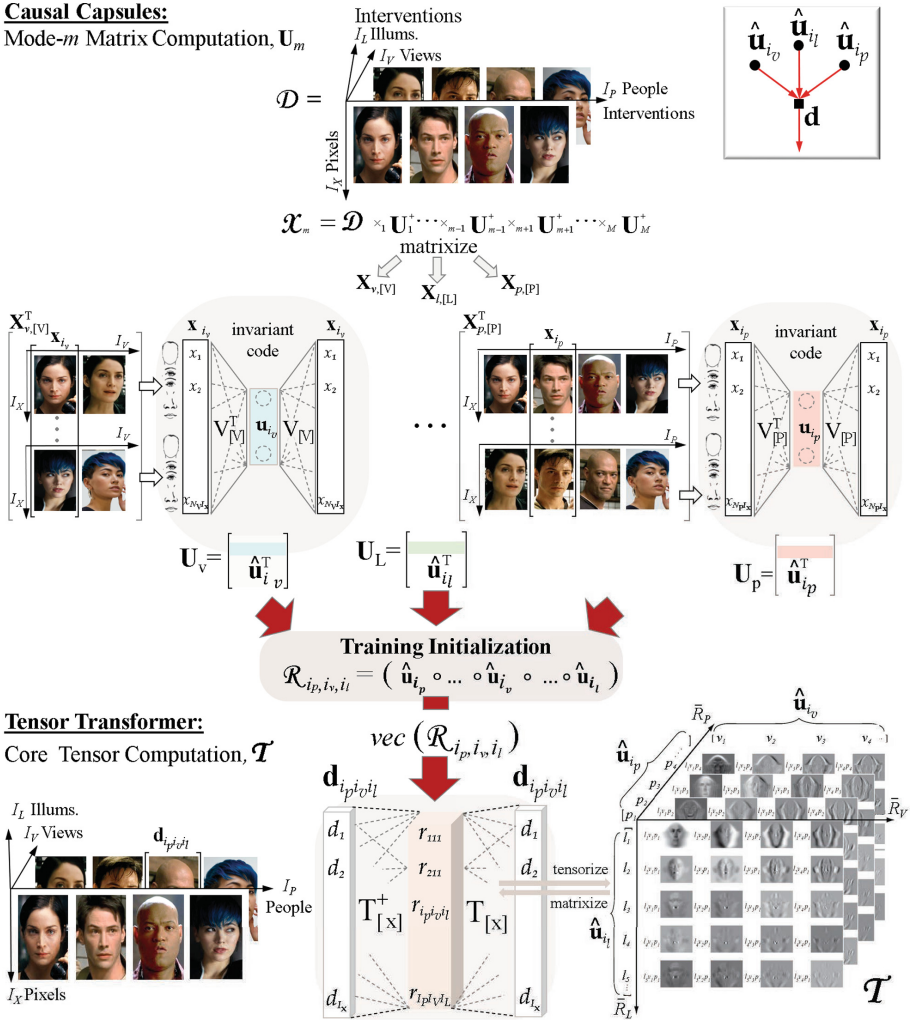
either experimental or observational studies, the statement “the effect of A is B” stays true regardless of new discoveries and changes in knowledge [29]. By comparison, interpretations are an understanding relative to a reference frame or a point of view. As new knowledge emerges, interpretations may be deemed to be inaccurate or invalid, which can undermine their reliability and usefulness in the development of trustworthy artificial intelligence. Interpretations are subject to reinterpretation.

<sup>2</sup> TensorFaces is a gentle introduction to causal tensor factor analysis [76, 79].



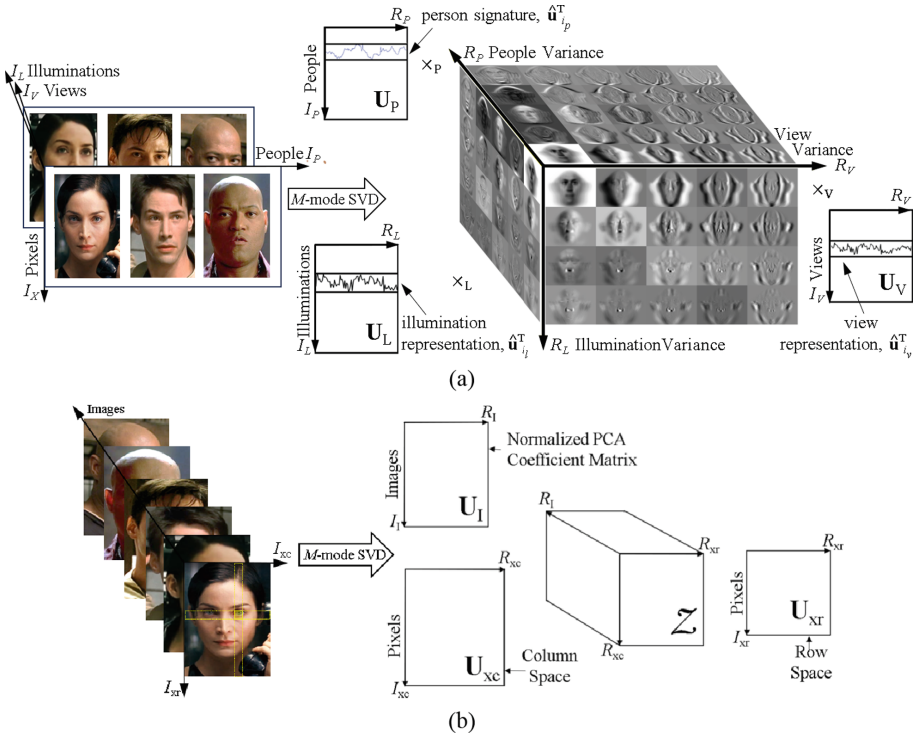
### 1.2 Causal Neural Networks

Causal neural networks are composed of causal capsules and tensor transformers, Fig. 1. *Causal capsules* estimate the latent variables that represent the causal factors of data formation, and a *tensor transformer* governs their interaction.



**Fig. 1.** A forward causal neural network is composed of a set of causal capsules and a tensor transformer. During training, causal capsules compute invariant causal factor representations,  $\hat{\mathbf{u}}_{im}$ , and the tensor transformer computes the extended core,  $\mathcal{T}$ , which governs their interaction. A tensor decoder with an estimated  $\mathbf{T}_{[X]}$ , is a generative model that computes the effects of new interventions. For scalable computation, each shallow autoencoder-decoder is replaced by a mathematically equivalent deep network, Fig. 3. In practice, images are vectorized and centered.

Causal capsules may be shallow autoencoder-decoder architectures that employ linear neurons and compute a set of invariant representations [1, 52, 58, 59, 61], as detailed in Supplemental A. The tensor transformer may be a *tensor autoencoder-decoder*, a shallow autoencoder-decoder whose code is the tensor product of the latent variables.



**Fig. 2.** Same data, same algorithm, but two different model specifications (problem setups). (a) Causal Inference: The  $M$ -mode SVD (Alg. 1) factorizes a “data tensor” of vectorized observations into a set of latent variables that represent the causal factors. (b) Simple regression: The  $M$ -mode SVD factorizes a “data tensor” composed of images as a “data matrix” into a column and row space, as well as into normalized PCA coefficients. (Images are vectorized except in Fig. 2b.)

Causal deep neural networks are composed by stacking autoencoders-decoders. Each autoencoder-decoder in a shallow causal neural network is replaced by mathematically equivalent deep neural network architectures that are derived by taking advantage of block algebra. An interleaved hierarchy of kernel functions [60] serves as a pre-processor that warps the data manifold for optimal tensor factor analysis.

---

**Algorithm 1**  $M$ -mode SVD (parallel computation)[76, 77]

---

**Input**  $\mathcal{D} \in \mathbb{C}^{I_0 \times \dots \times I_M}$ , dimensions  $R_0, R_1 \dots R_m \dots R_M$

1. Initialize  $\mathbf{U}_m := \mathbf{I}$  or random matrix,  $0 \leq m \leq M$
2. Iterate until convergence

For  $m := 0, \dots, M$ ,

$$\mathcal{A} := \mathcal{D} \times_0 \mathbf{U}_0^T \times \dots \times_{m-1} \mathbf{U}_{m-1}^T \times_{m+1} \mathbf{U}_{m+1}^T \dots \times \mathbf{U}_M^T$$

Set  $\mathbf{U}_m$  to the  $\tilde{R}_m$  leading left-singular vectors of the SVD of  $\mathbf{X}_{[m]}$  or SVD of  $[\mathbf{X}_{[m]} \mathbf{X}_{[m]}^T]^{a, b}$

3. Set  $\mathcal{Z} := \mathcal{D} \times_0 \mathbf{U}_0^T \dots \times_m \mathbf{U}_m^T \dots \times_M \mathbf{U}_M^T := \mathcal{A} \times \mathbf{U}_M^T$  <sup>c</sup>

**Output** mode matrices  $\mathbf{U}_0, \mathbf{U}_1, \dots, \mathbf{U}_M$  and core tensor  $\mathcal{Z}$ .

---

<sup>a</sup> The computation of  $\mathbf{U}_m$  in the SVD  $\mathbf{X}_{[m]} = \mathbf{U}_m \mathbf{\Sigma} \mathbf{V}_m^T$  can be performed efficiently, depending on which dimension of  $\mathbf{X}_{[m]}$  is smaller, by decomposing either  $\mathbf{X}_{[m]} \mathbf{X}_{[m]}^T = \mathbf{U}_m \mathbf{\Sigma}^2 \mathbf{U}_m^T$  (note that  $\mathbf{V}_m^T = \mathbf{\Sigma}^+ \mathbf{U}_m^T \mathbf{X}_{[m]}$ ) or by decomposing  $\mathbf{X}_{[m]}^T \mathbf{X}_{[m]} = \mathbf{V}_m \mathbf{\Sigma}^2 \mathbf{V}_m^T$  and then computing  $\mathbf{U}_m = \mathbf{X}_{[m]} \mathbf{V}_m \mathbf{\Sigma}^+$ .

<sup>b</sup> For a neural network implementation, the SVD of  $\mathbf{X}_{[m]}$  is replaced with an autoencoder that sequentially computes the orthonormal columns of  $\mathbf{U}_m / \mathbf{V}_m$  by performing gradient descent with the learning parameter  $\eta$  or stochastic gradient descent [9][56]. In Fig. 1, the autoencoders compute the columns of  $\mathbf{V}_m$ , where  $\mathbf{v}_{m,r}$  is the  $r$ th column and it represents the weights of the  $r$ th neuron. Matrix  $\mathbf{V}_{m,r}$  contains the first  $r$  columns of  $\mathbf{V}_m$ .

$$\left. \begin{array}{l} \text{Autoencoder} \\ \left\{ \begin{array}{l} \text{For } r := 1 \dots R_m. \\ \text{Iterate until convergence} \\ \Delta \mathbf{v}_{m,r}(t+1) = \eta \left( \mathbf{X}_{[m]} - \mathbf{V}_{m,r-1}(t) \mathbf{V}_{m,r-1}^T(t) \mathbf{X}_{[m]} \right) \underbrace{\mathbf{X}_{[m]}^T \mathbf{v}_{m,r}(t)}_{\text{code}} \\ \hat{\mathbf{v}}_{m,r}(t+1) = \frac{(\mathbf{v}_{m,r}(t) + \Delta \mathbf{v}_{m,r}(t+1))}{\|\mathbf{v}_{m,r}(t) + \Delta \mathbf{v}_{m,r}(t+1)\|} \end{array} \right. \end{array} \right\}$$

<sup>c</sup> The columns in  $\mathbf{Z}_{[0]}$  may be computed by initializing the code of an autoencoder to  $(\mathbf{U}_M \dots \otimes \mathbf{U}_m \dots \otimes \mathbf{U}_0)$ , where  $\otimes$  is the Kronecker product. In Fig. 1, the columns of the extended core  $\mathcal{T}$  are computed by initializing the code of the autoencoder with  $(\mathbf{U}_M \dots \otimes \mathbf{U}_m \dots \otimes \mathbf{U}_1)^T$  for batch training, and  $(\hat{\mathbf{u}}_{i_M}^T \dots \otimes \hat{\mathbf{u}}_{i_m}^T \dots \hat{\mathbf{u}}_{i_1}^T)^T$  when training one observation,  $\mathbf{d}_{i_1, \dots, i_M}$ , at a time.

---

A part-based deep neural network mirrors a part-based hierarchy of tensor factor models [73, 74][71, Sec 4.4], Supplemental C.<sup>3</sup>

Inverse causal neural networks implement the multilinear projection algorithm to estimate the causes of effects [72, 80]. A neural network that addresses an under-determined inverse problem is characterized by a wide hidden layer. Dimensionality reduction removes noise and nuisance variables [28, 66], and has the added benefit of reducing the widths of hidden layers. However, aggressive bottleneck dimension-

<sup>3</sup> There have been a number of related transformer architectures engineered and empirically tested with success [22, 47, 85].

ality reduction may camouflage an inherently ill-posed problem. Alternatively or in addition to dimensionality reduction and regularized regression, we prescribe modeling different aspects of the data formation process with piecewise tensor (multi-linear) models that return a set of candidate solutions [75]. Candidate solutions are gated to yield the most likely solution.

## 2 Forward Causal Question: “What If?”

Forward causal inference is a hypothesis-driven process that addresses the “what if” question. What if A is changed by one unit, what is the expected change in B? Causal hypotheses drive both the model specification and the experimental design for acquiring or generating training data.

*Training Data:* For modeling unit level effects of causes, the training data is generated by combinatorially varying each causal factor while holding the other factors fixed. The best causal evidence comes from randomized experimental studies. When randomized experiments for generating training data are unethical or infeasible, experimental studies may be approximated with carefully designed observational studies [57], such as natural experiments [2, 12, 36] or by employing the concept of transportability where learned causal effects from a set of experimental and observational studies are transferred to a new population, in which only observational studies can be conducted [55].<sup>4</sup>

### 2.1 Tensor Factor Analysis Model

Within the tensor mathematical framework (Supplemental Section B) a “data tensor,”  $\mathcal{D} \in \mathbb{C}^{I_0 \times I_1 \cdots \times I_m \cdots \times I_M}$ , contains a collection of vectorized<sup>5</sup> and centered observations,  $\mathbf{d}_{i_1, \dots, i_m, \dots, i_M} \in \mathbb{C}^{I_0}$  that are the result of  $M$  causal factors. Causal factor  $m$  ( $1 \leq m \leq M$ ) takes one of  $I_m$  values that are indexed by  $i_m$ ,  $1 \leq i_m \leq I_m$ . An observation and a data tensor may be modeled by a multilinear (tensor) principal component analysis (MPCA) equation

$$\begin{aligned} \mathcal{D} &= \mathcal{T} \times_1 \mathbf{U}_1 \cdots \times \mathbf{U}_m \times_M \mathbf{U}_M + \mathcal{E}, \\ \mathbf{d}_{i_1, \dots, i_M} &= \mathcal{T} \times_1 (\hat{\mathbf{u}}_{i_1}^T + \epsilon_{i_1}^T) \cdots \times_M (\hat{\mathbf{u}}_{i_M}^T + \epsilon_{i_M}^T) + \xi_{i_1, \dots, i_M}, \end{aligned} \quad (1)$$

where  $\mathcal{T}$  is the extended core that contains the basis vectors and governs the interaction between the latent variables  $\hat{\mathbf{u}}_{i_m}^T$  (row vector  $i_m$  of  $\mathbf{U}_m$ ), that represent the causal factors of data formation,  $\epsilon_{i_m} \in \mathcal{N}(\mathbf{0}, \Sigma_m)$  are disturbances with Gaussian

<sup>4</sup> Datasheets for datasets, as proposed by Gebru *et al.* [24], may help facilitate the approximation of experimental studies.

<sup>5</sup> It is preferable to vectorize an image and treat it as a single observation rather than as a collection of independent column/row observations. Most assertions found in highly cited publications in favor of treating an image as a “data matrix” or “tensor” do not stand up to analytical scrutiny [71, App. A].

distribution, and  $\xi_{i_1, \dots, i_M}$  is a Gaussian measurement error. Minimizing the cost function

$$L = \|\mathcal{D} - \mathcal{T} \times_1 \mathbf{U}_1 \dots \times_m \mathbf{U}_m \dots \times_M \mathbf{U}_M\| + \sum_{m=1}^M \lambda_m \|\mathbf{U}_m^T \times_1 \mathbf{U}_m - \mathbf{I}\| \quad (2)$$

is equivalent to maximum likelihood estimation [19] of the causal factor parameters, assuming the data was generated by the model with additive Gaussian noise. The mode matrices  $\mathbf{U}_m$  are computed by employing a set of  $M$  alternating least squares optimizations,

$$L_m = \|\mathcal{X}_m - \mathcal{T} \times_m \mathbf{U}_m\| + \lambda_m \|\mathbf{U}_m^T \times_1 \mathbf{U}_m - \mathbf{I}\|, \quad (3)$$

where

$$\mathcal{X}_m := \mathcal{D} \times_1 \dots \times_{m-1} \mathbf{U}_{m-1}^T \times_{m+1} \mathbf{U}_{m+1}^T \dots \times_M \mathbf{U}_M^T \quad \text{- parallel computation} \quad (4)$$

$$= \mathcal{X}_m(t-1) \times_n \mathbf{U}_n^T(t) \mathbf{U}_n(t-1), \quad \forall n \neq m, \quad \text{- asynchronous parallel computation} \quad (5)$$

$$= (\mathcal{X}_{m-1} \times_{m-1} \mathbf{U}_{m-1}^T) \times_m \mathbf{U}_m \quad \text{- sequential computation} \quad (6)$$

$$= \mathcal{T} \times_m \mathbf{U}_m \quad (7)$$

The  $M$ -mode SVD [76] (Alg. 1) minimizes the  $M$  alternating least squares (3) in closed form by employing  $M$  different SVDs. The approach is suitable for parallel (4), asynchronous (5), or sequential (6) computation. The extended core tensor  $\mathcal{T}$  is computed by multiplying the data tensor with the inverse mode matrices,  $\mathcal{T} = \mathcal{D} \times_1 \mathbf{U}^T_1 \dots \times_m \mathbf{U}^T_m \dots \times_M \mathbf{U}^T_M$ , or more efficiently as  $\mathcal{T} = \mathcal{X}_m \times \mathbf{U}_m^T$ .

### 2.2 Kernel Tensor Factor Analysis Model

When data  $\mathcal{D}$  are a tensor combination  $\phi(\mathcal{T})$  of non-linear independent causal factors  $\phi_m(\mathbf{C}_m)$ . Kernel multilinear independent component analysis (K-MICA) [71, Ch 4.4] employs the “kernel trick” [60, 82] as a pre-processing step which makes the data suitable for multilinear independent component analysis [79] (Alg. 2),

$$\begin{aligned} \mathcal{D} &= \mathcal{T} \times_1 \mathbf{C}_1 \dots \times_m \mathbf{C}_m \dots \times_M \mathbf{C}_M + \mathcal{E} \\ \mathbf{C}_m &= \mathbf{U}_m \mathbf{W}^{-1} + \mathbf{E}_m, \end{aligned} \quad (8)$$

based on negentropy, mutual information, or higher-order cumulants. K-MPCA is a tensor generalization of the kernel PCA [60] and K-MICA is a tensor generalization of kernel ICA [3, 88].

To accomplish this analysis, recall that the computation of covariance matrix  $\mathbf{D}_{[m]} \mathbf{D}_{[m]}^T$  involves inner products  $\mathbf{d}_{i_1 \dots i_{m-1} j}^T \mathbf{d}_{i_2 \dots i_{m-1} k}$  between pairs of data points in the data tensor  $\mathcal{D}$  associated with causal factor mode  $m$ , for  $m = 1, \dots, M$  (Step 2.2 in Algorithm 1). We replace the inner products with a generalized distance measure between images,  $K(\mathbf{d}_{i_1 \dots i_{m-1} j}, \mathbf{d}_{i_2 \dots i_{m-1} k})$ , where  $K(\cdot, \cdot)$  is a suitable kernel function (Table 1) that corresponds to an inner product in some

**Algorithm 2.** Kernel Tensor Factor Analysis [71, Sec 4.4][79]  
 Kernel Multilinear Independent Component Analysis (K-MICA) and  
 Kernel Principal Component Analysis (K-MPCA).

**Input** the data tensor  $\mathcal{D} \in \mathbb{C}^{I_0 \times \dots \times I_M}$ , where mode  $m = 0$  is the measurement mode, and the desired ranks are  $\tilde{R}_1, \dots, \tilde{R}_M$ .

Initialize  $\mathbf{C}_m = \mathbf{I}, \forall 0 \leq m \leq M$

Iterate until convergence.

1. For  $m := 1, \dots, M$

(a) Set  $\mathcal{X}_m := \mathcal{D} \times_1 \mathbf{C}_1^+ \cdots \times_{m-1} \mathbf{C}_{m-1}^+ \times_{m+1} \mathbf{C}_{m+1}^+ \cdots \times_M \mathbf{C}_M^+$ .

(b) Compute the elements of the mode- $m$  covariance matrix using kernel functions, Table 1, for  $j, k := 1, \dots, I_m$ :

$$[\mathbf{x}_{m[m]} \mathbf{x}_{m[m]}^T]_{jk} := \sum_{i_1=1}^{I_1} \cdots \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_{m+1}=1}^{I_{m+1}} \cdots \sum_{i_M=1}^{I_M} K(\mathbf{x}_{i_1 \dots i_{m-1} j i_{m+1} \dots i_M}, \mathbf{x}_{i_1 \dots i_{m-1} k i_{m+1} \dots i_M}). \quad (9)$$

(c) <sup>a</sup>  $\left\{ \begin{array}{l} \text{For K-MPCA: Set } \mathbf{C}_m := \mathbf{U}, \text{ the left matrix of the SVD of } [\mathbf{X}_{[m]} \mathbf{X}_{[m]}^T] \text{ from (9)} \\ \text{Truncate to } \tilde{R}_m \text{ columns } \mathbf{U}_m \in \mathbb{C}^{I_m \times \tilde{R}_m}. \\ \text{For K-MICA: Set } \mathbf{C}_m := \mathbf{U}_m \mathbf{W}_m^{-1}. \text{ The additional rotation matrix } \mathbf{W}_m \text{ may be} \\ \text{computed based on negentropy, mutual information, or higher-} \\ \text{order cumulants [79]. The initial SVD of } [\mathbf{X}_{[m]} \mathbf{X}_{[m]}^T] \text{ from (9)} \\ \text{truncates the subspace to } \tilde{R}_m. \end{array} \right.$

2. Set  $\mathcal{T} := \mathcal{X}_M \times_M \mathbf{C}_M^+$ . For K-MPCA,  $\mathbf{C}_M^+ = \mathbf{C}_M^T$ .

**Output** the converged extended core tensor  $\mathcal{T} \in \mathbb{C}^{I_0 \times \tilde{R}_1 \times \dots \times \tilde{R}_M}$  and causal factor mode matrices  $\mathbf{C}_1, \dots, \mathbf{C}_M$ .

<sup>a</sup> Every SVD step may be autoencoder-decoder. See Algorithm 1, Footnotes *a* and *b*.

See Fig. 3 for a scalable neural network implementation. expanded feature space. This generalization naturally leads us to a *Kernel Multilinear PCA (K-MPCA) Algorithm*, where the covariance computation is replaced by

$$[\mathbf{D}_{[m]} \mathbf{D}_{[m]}^T]_{jk} := \sum_{i_1=1}^{I_1} \cdots \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_{m+1}=1}^{I_{m+1}} \cdots \sum_{i_M=1}^{I_M} K(\mathbf{d}_{i_1 \dots i_{m-1} j i_{m+1} \dots i_M}, \mathbf{d}_{i_1 \dots i_{m-1} k i_{m+1} \dots i_M}).$$

When a causal factor is a combination of multiple independent sources that are causal in nature, we employ a rotation matrix  $\mathbf{W}$  to identify them. The rotation matrix is computed by employing either mutual information, negentropy, or higher-order cumulants [4, 5, 16, 34]. A *Kernel Multilinear ICA (K-MICA) Algorithm* is a kernel generalization of the multilinear independent component analysis (MICA) algorithm [79]. Algorithm 2 simultaneously specifies both K-MPCA and K-MICA algorithms. A scalable tensor factor analysis represents an observation as a hierarchy of parts and wholes [73, 74].

### 2.3 Neural Network Architecture

Tensor factor analysis models are transformed into causal neural networks by using autoencoder-decoders as building blocks. Causal neural networks are composed

**Table 1.** Common kernel functions. Kernel functions are symmetric, positive semi-definite functions corresponding to symmetric, positive semi-definite Gram matrices. The linear kernel does not modify or warp the feature space.

Linear kernel:	$K(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} = \mathbf{u} \cdot \mathbf{v}$
Polynomial kernel of degree $d$ :	$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v})^d$
Polynomial kernel up to degree $d$ :	$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v} + 1)^d$
Sigmoidal kernel:	$K(\mathbf{u}, \mathbf{v}) = \tanh(\alpha \mathbf{u}^T \mathbf{v} + \beta)$
Gaussian (radial basis function (RBF)) kernel:	$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\ \mathbf{u}-\mathbf{v}\ ^2}{2\sigma^2}\right)$

of causal capsules and tensor transformers, Fig. 1. *Causal capsules* estimate a set of latent variables that represent the causal factors of data formation. A *tensor transformer* governs the causal factor interaction. The M-mode SVD (Algorithm 1) is transformed into a neural network by replacing every SVD step with gradient descent optimization, which is outsourced to an autoencoder-decoder with neurons that have a linear transfer function, Supplemental A. For effectiveness, we employ stochastic gradient descent [9, 56]. The extended core tensor  $\mathbf{T}_{[0]}$  is computed by defining and employing a tensor autoencoder, an autoencoder whose code is initialized to the tensor product of the causal factor representations,  $\{\mathbf{u}_{i_m} | 1 \leq i_m \leq I_m \text{ and } 1 \leq m \leq M\}$ ,

$$\mathbf{d}_{i_1, \dots, i_m \dots i_M} = \mathbf{T}_{[0]}(\mathbf{u}_{i_M}^T \otimes \dots \otimes \mathbf{u}_{i_m}^T \dots \otimes \mathbf{u}_{i_1}^T)^T.$$

To address a set of arbitrarily non-linear causal factors, each autoencoder employs kernel functions (Table 1).

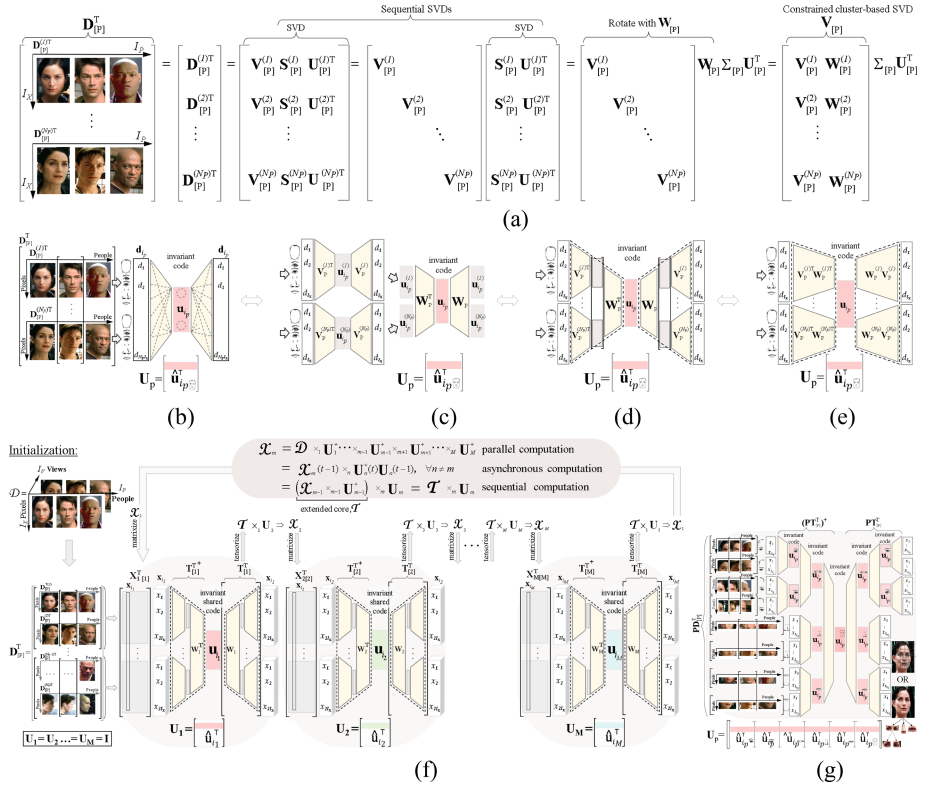
## 2.4 Causal Deep Networks and Scalable Tensor Factor Analysis:

For a scalable architecture, we leverage the properties of block algebra. Shallow autoencoders are replaced with either a mathematically equivalent deep neural network that is a part-based hierarchy of autoencoders-decoders, or a set of concurrent autoencoders-decoders, Fig. 3.

For example, the orthonormal subspace of a data batch,  $\mathbf{D} \in \mathbb{C}^{I_0 \times I_1}$  that has  $I_0$  measurements and  $I_1$  observations may be computed by recursively subdividing the data and analyzing the data blocks,

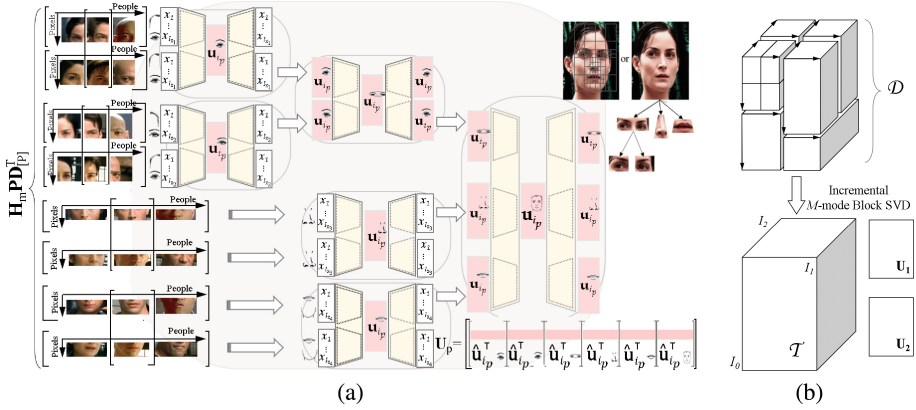
$$\begin{aligned} \mathbf{D} &= \begin{bmatrix} \mathbf{D}_A \\ \mathbf{D}_B \end{bmatrix} = \begin{bmatrix} \mathbf{U}_A \mathbf{S}_A \mathbf{V}_A^T \\ \mathbf{U}_B \mathbf{S}_B \mathbf{V}_B^T \end{bmatrix} = \begin{bmatrix} \mathbf{U}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_B \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{S}_A \mathbf{V}_A^T \\ \mathbf{S}_B \mathbf{V}_B^T \end{bmatrix}}_{\text{SVD}} = \\ &= \begin{bmatrix} \mathbf{U}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_B \end{bmatrix} \mathbf{W} \mathbf{\Sigma} \mathbf{V}^T = \begin{bmatrix} \mathbf{U}_A \mathbf{W}_A \\ \mathbf{U}_B \mathbf{W}_B \end{bmatrix} \mathbf{\Sigma} \mathbf{V}^T = \\ &= \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \end{aligned} \tag{9}$$

where  $\mathbf{W}$  is a rotation matrix that transforms the basis matrices,  $\mathbf{U}_A$  and  $\mathbf{U}_B$ , spanning the observations in data blocks,  $\mathbf{D}_A$  and  $\mathbf{D}_B$ , such that their observations no



**Fig. 3.** Deep neural network. Subfigures (a-e) depict [10–14] [71, pg.38–40]. (a) The mode matrix computation  $\mathbf{U}_m$  may be thought as a constrained cluster-based PCA that is rewritten in terms of block SVDs. Matrixizing may be viewed as a concatenation of “cluster” data. The matrix  $\mathbf{W}$  transforms the basis matrix  $\mathbf{V}_0^{(r)}$  such that the causal factor representation  $\mathbf{U}_m$  is the same regardless of cluster membership. In a tensor model, there are  $M$  different constrained cluster-based PCAs. (b) Mode matrix  $\mathbf{U}_m$  computation using a single autoencoder-decoder. (c) Mode matrix computation as a hierarchy of autoencoder-decoders, (d) Mode matrix computation written as a deep learning model (e) Concurrent-autoencoders; *i.e.*, constrained cluster-based autoencoders. (f) Forward causal model with a set of capsules implemented by deep neural networks. For a parallel, synchronized or asynchronous computation, we break the chain links and shuttle causal information,  $\mathbf{U}_m$ , between capsules to compute  $\mathcal{X}(t+1)$  for the next iteration. (g) Each capsule in (f) may be replaced with a part-based deep neural network by permuting the rows in  $\mathbf{D}_{[P]}^{(j)}$  with  $\mathbf{P}$ , segmented by  $\mathbf{H}_m$ , which is efficiently trained with a part-based hierarchy of autoencoders, Fig. 4.





**Fig. 4.** (a) Causal capsules may be implemented with a part-based hierarchy of autoencoders. The dataset is permuted by  $\mathbf{P}$ , segmented and filtered by  $\mathbf{H}_m$  that is mode dependent. (b) Implementing the capsules with a part-based hierarchy of autoencoders is equivalent to performing M-mode Block SVD [70, 74, Sec IV]

longer have distinct representations  $\mathbf{V}_A^T$  and  $\mathbf{V}_B^T$ , but have the same representations  $\mathbf{V}^T$ .<sup>6</sup>

Computing causal factor representations, the mode matrices  $\mathbf{U}_m$  of an MPCA tensor model, is equivalent to computing a  $M$  different of mutually constrained, cluster-based PCA, Fig. 3a. When dealing with data that can be separated into clusters, the standard machine learning approach is to compute a separate PCA. When data from different clusters are generated by the same underlying process (e.g., facial images of the same person under different viewing conditions), the data blocks can be concatenated in the measurement mode and the common causal factor can be modeled by one PCA. However, for a scalable solution, we employ block algebra (10) and compute a set of constrained cluster-based PCAs, *i.e.*, a set of concurrent PCAs.

Thus, we define a *constrained, cluster-based PCA* as the computation of a set of PCA basis vectors, such that the latent representation is constrained to be the invariant of the cluster membership.

In the context of our multifactor data analysis, we define a cluster as a set of observations for which all factors are fixed but one. For every tensor mode, there are  $N_m = I_1 I_2 \dots I_{m-1} I_{m+1} \dots I_M$  possible clusters and the data in each cluster varies with the same causal mode. The constrained, cluster-based PCA concatenates the clusters in the measurement mode and analyzes the data with a linear model, such as PCA.

To see this, let  $\mathcal{D}_{i_1 \dots i_{m-1} i_{m+1} \dots i_M} \in \mathbb{C}^{I_0 \times 1 \times 1 \dots \times 1 \times I_m \times 1 \times 1 \dots \times 1}$  denote a subtensor of  $\mathcal{D}$  that is obtained by fixing all causal factor modes but mode  $m$  and mode 0

<sup>6</sup> Block algebra may be employed if the tensor model is multilinear (tensor) principal component analysis (MPCA), multilinear (tensor) independent component analysis (MICA) [79], Kernel-MPCA or Kernel-MICA [71].

(the measurement mode). Matrixizing this subtensor in the measurement mode we obtain  $\mathbf{D}_{i_1 \dots i_{m-1} i_{m+1} \dots i_M [0]} \in \mathbb{C}^{I_0 \times I_m}$ . This data matrix comprises a cluster of data obtained by varying causal factor  $m$ , to which one can traditionally apply PCA. Since there are  $N_m = I_1 I_2 \dots I_{m-1} I_{m+1} \dots I_M$  possible clusters that share the same underlying space associated with factor  $m$ , the data can be concatenated and PCA performed in order to extract the same representation for factor  $m$  regardless of the cluster. Now, consider the MPCA computation of mode matrix  $\mathbf{U}_m$ , Fig. 3a, which can be written in terms of matrixized subtensors as

$$\mathbf{D}_m = \begin{bmatrix} \mathbf{D}_{1 \dots 11 \dots 1 [m]}^T \\ \vdots \\ \mathbf{D}_{I_1 \dots 11 \dots 1 [m]}^T \\ \vdots \\ \mathbf{D}_{I_1 \dots I_{m-1} I_{m+1} \dots I_M [m]}^T \end{bmatrix}^T = \mathbf{U}_m \mathbf{\Sigma}_m \mathbf{V}_m^T. \quad (10)$$

This is equivalent to computing a set of  $N_m = I_1 I_2 \dots I_{m-1} I_{m+1} \dots I_M$  cluster-based PCAs concurrently by combining them into a single statistical model and representing the underlying causal factor  $m$  common to the clusters. Thus, rather than computing a separate linear PCA model for each cluster, MPCA concatenates the clusters into a single statistical model and computes a representation (coefficient vector) for mode  $m$  that is invariant relative to the other causal factor modes  $1, \dots, (m-1), (m+1), \dots, M$ . For a scalable solution, we rotate the cluster-based PCA basis vectors, such that the data blocks have the same representation regardless of cluster membership. Thus, MPCA is a multilinear, constrained, cluster-based PCA.

To clarify the relationship, let us number each of the matrices  $\mathbf{D}_{i_1 \dots i_{m-1} i_{m+1} \dots i_M [m]} = \mathbf{D}_m^{(n)}$  with a parenthetical superscript  $1 \leq n = 1 + \sum_{k=1, k \neq m}^M (i_k - 1) \prod_{l=1, l \neq m}^{k-1} I_l \leq N_m$ .

Let each of the cluster SVDs be  $\mathbf{D}_m^{(n)} = \mathbf{U}_m^{(n)} \mathbf{\Sigma}_m^{(n)} \mathbf{V}_m^{(n)T}$ , and

$$\mathbf{D}_{[m]} = \underbrace{[\mathbf{U}_m^{(1)} \mathbf{\Sigma}_m^{(1)} \dots \mathbf{U}_m^{(N_m)} \mathbf{\Sigma}_m^{(N_m)}]}_{\text{SVD}} \text{diag}([\mathbf{V}_m^{(1)} \dots \mathbf{V}_m^{(N_m)}])^T \quad (11)$$

$$= \mathbf{U}_m \mathbf{\Sigma}_m \mathbf{W}_m^T \text{diag}([\mathbf{V}_m^{(1)} \dots \mathbf{V}_m^{(N_m)}])^T, \quad (12)$$

$$= \mathbf{U}_m \mathbf{\Sigma}_m [\mathbf{V}_m^{(1)} \mathbf{W}_m^{(1)} \dots \mathbf{V}_m^{(N_m)} \mathbf{W}_m^{(N_m)}]^T \quad (13)$$

$$= \mathbf{U}_m \mathbf{\Sigma}_m \mathbf{V}_m^T, \quad (14)$$

where  $\text{diag}(\cdot)$  denotes a diagonal matrix whose elements are each of the elements of its vector argument. The mode matrix  $\mathbf{V}_m^{(n_m)}$  is the measurement matrix  $\mathbf{U}_0^{(n_m)}$  ( $\mathbf{U}_x^{(n_m)}$  when the measurements are image pixels) that contains the eigenvectors spanning the observed data in cluster  $n_m$ ,  $1 \leq n_m \leq N_m$ . MPCA can be thought as computing a rotation matrix,  $\mathbf{W}_m$ , that contains a set of blocks  $\mathbf{W}_m^{(n)}$  along the diagonal that transform the PCA cluster eigenvectors  $\mathbf{V}_m^{(n_m)}$  such that the mode

matrix  $\mathbf{U}_m$  is the same regardless of cluster membership (11–14), Fig 3. The constrained “cluster”-based PCAs may also be implemented with a set of concurrent “cluster”-based PCAs, Fig. 3e.

Causal factors of object wholes may be computed efficiently from their parts, by applying a permutation matrix  $\mathbf{P}$  and creating part-based data clusters with a segmentation filter  $\mathbf{H}_m$ , where  $\mathcal{D}^{\text{T} \times_m} \mathbf{H}_m \mathbf{P} \Leftrightarrow \mathbf{H}_m \mathbf{P} \mathbf{D}_{[m]}^{\text{T}}$ , but leaving prior analysis intact, Fig. 3g. A deep neural network can be efficiently trained with a hierarchy of part-based autoencoders, Fig. 4. A computation that employs a part-based hierarchy of autoencoders parallels the Incremental M-mode Block SVD [70, 74, Sec. IV]. A data tensor is recursively subdivided into data blocks, analyzed in a bottom-up fashion, and the results merged as one moves through the hierarchy. The computational cost is the cost of training one autoencoder,  $\mathcal{O}(T)$ , times  $\mathcal{O}(\log N_M)$ , the total number of autoencoders trained for each factor matrix,  $\mathcal{O}(T \log N_m)$ . If the causal neural network is trained sequentially, the training cost for one time iteration is  $\mathcal{O}(MT \log \bar{N})$ , where  $\bar{N}$  is the average number of clusters across the  $M$  modes.

### 3 Inverse Causal Question: “Why?”

Inverse causal inference addresses the “why” question and estimates the causes of effects given an estimated forward causal model and a set of constraints that reduce the solution set<sup>7</sup> and render the problem well-posed [25, 72, 80].

Multilinear tensor factor analysis constrains causal factor representations to be unitary vectors. Multilinear projection [72, 80] relies on this constraint and performs multiple regularized regressions. One or more unlabeled test observations that are not part of the training data set are simultaneously projected into the causal factor spaces

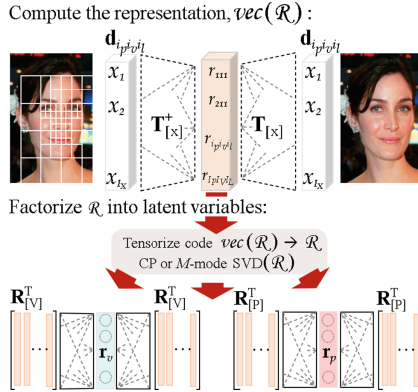
$$\begin{aligned} \mathbf{T}^{\text{T} \times_x} \times_x^{\text{T}} \mathbf{d}_{\text{test}} &= \mathcal{R} \text{ followed by } M\text{-mode SVD/CP of } \mathcal{R} \\ &\approx \mathbf{r}_1 \dots \circ \mathbf{r}_m \dots \circ \mathbf{r}_M, \text{ and } \|\mathbf{r}_m\| = 1. \end{aligned}$$

A neural network that implements a multilinear projection architecture is an inverted (upside down) forward neural network architecture that employs an estimated  $\mathbf{T}_{[x]}^+$  and reverses the operation order, Fig. 5.

Neural architectures addressing underdetermined inverse problems are characterized by hidden layers that are wider than the input layer; *i.e.*, the dimensionality of  $\text{vec}(\mathcal{R})$  is larger than the number of measurements in  $\mathbf{d}$ . Dimensionality reduction reduces noise, and the width of the hidden layers [28]. However, they can also camouflage an inherently underdetermined inverse problem. Adding sparsity, non-negativity constraints [81], etc., can further reduce the solution set in a principled way. Alternatively or in addition, one can determine a set of candidate solutions

<sup>7</sup> Different combinations of the same causal factors can lead to the same outcome. In imaging, these are known as visual illusions. This is a many-to-one problem, and its inverse is ill-posed without constraints.

by modeling different aspects of the mechanism of data formation as piecewise tensor (multilinear) factor models. A single multilinear projection [72, 80] is replaced with multiple multilinear projections. Vasilescu and Terzopoulos [75] rewrote the forward multilinear model in terms of multiple piecewise linear models that were employed to perform multiple linear projections and produced multiple candidate solutions that were gated to return the most likely solution.



**Fig. 5.** An inverse causal network is an inverted (upside-down) forward network that implements the multilinear projection [72, 80]. Operations are performed in reverse order using the estimated  $\mathbf{T}_{[X]}^+$  from the forward pass. For a scalable solution, autoencoder-decoders are replaced with a deep network, Fig. 3

## 4 Conclusion

We derive a set of shallow and deep causal neural networks that are a consequence of causal tensor factor analysis and block algebra. Causal neural networks are composed of causal capsules and a tensor transformer. Causal capsules compute invariant causal factor representations, whose interaction are governed by a tensor transformation. An inverse causal neural network estimates the causes of effects and implements the multilinear projection. As an alternative to aggressive “bottle-neck” dimensionality reduction that may camouflage an inherently underdetermined inverse problem, the mechanism of data formation is modeled as piecewise tensor (multilinear) models, and inverse causal neural networks perform multiple multilinear projections that result in multiple candidate solutions, which may be gated to yield the most likely solution.

## References

1. Ackley, D.H., Hinton, G.A., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. *Cogn. Sci.* **9**(1), 147–169 (1985)

2. Angrist, J.D., Lifetime earnings and the Vietnam era draft lottery: evidence from social security administrative records. *American Econ. Rev.* 313–336 (1990)
3. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. *J. Mach. Learn. Res.* **3**(Jul), 1–48 (2002)
4. Bartlett, M., Movellan, J., Sejnowski, T.: Face recognition by independent component analysis. *IEEE Trans. Neural Networks* **13**(6), 1450–64 (2002)
5. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* **6**, 1004–1034 (1995)
6. Benesty, J., Paleologu, C., Dogariu, L., Ciocină, S.: Identification of linear and bilinear systems: a unified study. *Electronics* **10**(15) (2021)
7. Bengio, Y., Courville, A.: *Handbook on neural information processing*, chapter Deep Learning of Representations, pp. 1–28. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
8. Bentler, P., Lee, S.: A statistical development of three-mode factor analysis. *British J. Math. and Stat. Psych.* **32**(1), 87–104 (1979)
9. Bottou, L., et al.: Online learning and stochastic approximations. *On-line Learn. Neural Netw.* **17**(9), 142 (1998)
10. Bro, R.: Parafac: tutorial and applications. *Chemom. Intell. Lab. Syst. Special Issue 2nd Internet Cont. in Chemometrics (INCINC'96)* **38**(2), 149–171 (1997)
11. Bulat, A., Kossai, J., Tzimiropoulos, G., Pantic, M.: Incremental multi-domain learning with network latent tensor factorization. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pp. 10470–10477. AAAI Press (2020)
12. Card, D., Krueger, A.B.: Minimum wages and employment: a case study of the fast food industry in New Jersey and Pennsylvania (1993)
13. Carroll, J.D., Chang, J.J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika* **35**, 283–319 (1970)
14. Chen, J.C., Ranjan, R., Kumar, A., Chen, C.H., Patel, V.M., Chellappa, R.: An end-to-end system for unconstrained face verification with deep convolutional neural networks. In: *IEEE International conference on Computer Vision Workshop (ICCVW)*, pp. 360–368 (2015)
15. Chu, W., Ghahramani, Z.: Probabilistic models for incomplete multi-dimensional arrays. In: *Artificial Intelligence and Statistics of Proceedings of Machine Learning Research vol. 5*, pp. 89–96, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 (2009). PMLR
16. Common, P.: Independent component analysis, a new concept? *Signal process.* **36**, 287–314 (1994)
17. Davis, J., Gao, H.: Recognizing human action efforts: an adaptive three-mode PCA framework. In: *Proceedings IEEE International conference on Computer Vision, (ICCV)*, pp. 1463–1469 Nice, France (2003)
18. de Lathauwer, L., de Moor, B., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. of Matrix Anal. Appl.* **21**(4), 1253–1278 (2000)
19. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **39**(1), 1–22 (1977)
20. Desjardins, G., Courville, A., Bengio, Y.: Disentangling factors of variation via generative entangling. [arXiv:1210.5474](https://arxiv.org/abs/1210.5474) (2012)
21. Dudek, J.M., Dueñas-Osorio, L., Vardi, M.Y.: Efficient contraction of large tensor networks for weighted model counting through graph decompositions (2019)
22. Fan, H., et al.: Multiscale vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6824–6835 (2021)

23. Fraassen, B.C.V.: *The Scientific Image*, Oxford University Press, 12 (1980)
24. Gebru, T., et al.: Datasheets for datasets. *Commun. ACM* **64**(12), 86–92 (2021)
25. Gelman, A., Imbens, G.: Why ask why? Forward causal inference and reverse causal questions. Tech. report, Nat. Bureau of Econ. Research (2013)
26. Grindlay, G., Vasilescu, M.A.O.: A multilinear (tensor) framework for HRTF analysis and synthesis. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07, vol. 1, pp. 1–161–164 (2007)
27. Harshman, R.: Foundations of the PARAFAC procedure: model and conditions for an explanatory factor analysis. Tech. Report Working Papers Phonetics 16, UCLA, CA (1970)
28. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
29. Holland, P.W.: Statistics and causal inference: rejoinder. *J. of the American Stat. Assoc.* **81**(396), 968–970 (1986)
30. Hoover, R.C., Caudle, K., Braman, K.: A new approach to multilinear dynamical systems and control (2021)
31. Hsu, E., Pulli, K., Popovic, J.: Style translation for human motion. *ACM Trans. Graphics* **24**(3), 1082–89 (2005)
32. Huang, G.B.: Learning hierarchical representations for face verification with convolutional deep belief networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2518–2525 (2012)
33. Wang, H., Ahuja, N.: Facial expression decomposition. In: Proceedings 9th IEEE International Conference on Computer Vision (ICCV), pp. 958–965 (2003)
34. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. Wiley, New York (2001)
35. Imbens, G., Rubin, D.: *Causal Inference for Statistics, Social and Biomedical Sciences: An Introduction*. Cambridge Univ. Press (2015)
36. Imbens, G.W., Angrist, J.D.: Identification and estimation of local average treatment effects. *Econometrica* **62**(2), 467–475 (1994)
37. Iwen, M.A., Needell, D., Rebrova, E., Zare, A.: Lower memory oblivious (tensor) subspace embeddings with fewer random bits: modewise methods for least squares. *SIAM J. Matrix Anal. Appl.* **42**(1), 376–416 (2021)
38. Kapteyn, A., Neudecker, H., Wansbeek, T.: An approach to  $n$ -mode component analysis. *Psychometrika* **51**(2), 269–275 (1986)
39. Kermany, D.S., et al.: Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* **172**(5), 1122–1131.e9 (2018)
40. Khrulkov, V.: *Geometrical Methods in Machine Learning and Tensor Analysis*, PhD dissertation, Skolkovo Institute (2020)
41. Kim, Y., Park, E., Yoo, S., Choi, T., Yang, L., Shin, D.: Compression of deep convolutional neural networks for fast and low power mobile applications. *CoRR*, abs/1511.06530 (2015)
42. Kossaifi, J., Lipton, Z.C., Kolbeinsson, A., Khanna, A., Furlanello, T., Anandkumar, A.: Tensor regression networks. *J. Mach. Learn. Res.* **21**(123), 1–21 (2020)
43. Kroonenberg, P.M., de Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika* **45**, 69–97 (1980)
44. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I.V., Lempitsky, V.S.: Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *CoRR*, abs/1412.6553 (2014)
45. Leviton, L.: External validity. In: Smelser, N.J., Baltes, P.B. (eds.) *International Encyclopedia of the Social & Behavioral Sciences*, pp. 5195–5200. Pergamon, Oxford (2001)

46. Lewis, D.: Causal Explanation. In: Philosophical Papers vol. II, pp. 214–240 Oxford University Press (1987)
47. Liu, Z.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
48. Madani, A., Moradi, M., Karargyris, A., Syeda-Mahmood, T.: Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 1038–1042 (2018)
49. Magnus, J., Neudecker, H.: Matrix Differential Calculus with Applications in Statistics and Econometrics. Wiley (1988)
50. Memisevic, R., Hinton, G.E.: Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Comput.* **22**(6), 1473–1492 (2010)
51. Novikov, A., Podoprikhin, D., Osokin, A., Vetrov, D.P.: Tensorizing neural networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., eds, *Advances in Neural Information Processing Systems 28*, pp. 442–450. Curran Associates, Inc (2015)
52. Oja, E.: A simplified neuron model as a principal component analyzer. **15**, 267–2735 (1982)
53. Onu, C.C., Miller, J.E., Precup, D.: A fully tensorized recurrent neural network. *CoRR*, abs/2010.04196 (2020)
54. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge Univ, Press (2000)
55. Pearl, J., Bareinboim, E.: External validity: from do-calculus to transportability across populations. *Stat. Sci.* **29**(4), 579–95 (2014)
56. Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* 400–407 (1951)
57. Rubin, D.B.: For objective causal inference, design trumps experimental analysis. *Ann. Appl. Stat.* **2**(3), 808–840 (2008)
58. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation (1986)
59. Sanger, T.: Optimal unsupervised learnig in a single layer linear feedforward neural network. **12**, 459–473 (1989)
60. Schölkoph, B., Smola, A., Muller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**(5), 1299–1319 (1998)
61. Sejnowski, T., Chattarji, S., Sfanton, P.: Induction of synaptic plasticity by hebbian covariance in the hippocampus In: *The Computing Neuron*, pp. 105–124. Addison-Wesley (1989)
62. Spirtes, P., Glymour, C.N., Scheines, R., Heckerman, D.: Causation, prediction, and search, MIT press (2000)
63. Sun, Y., Wang, X., Tang, X.: Hybrid deep learning for face verification. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1489–96 (2013)
64. Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–08 (2014)
65. Tang, Y., Salakhutdinov, R., Hinton, G.: Tensor analyzers. In: Proceedings of Machine Learning Research, vol. 28 pp. 163–171, Atlanta, Georgia, USA 17–19 Jun (2013)
66. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: 2015 IEEE Information Theory Workshop (ITW), pp. 1–5 (2015)

67. Topol, E.J.: High-performance medicine: the convergence of human and artificial intelligence. *Nat. Med.* **25**(1), 44–56 (2019)
68. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. *Psychometrika* **31**, 279–311 (1966)
69. Vasilescu, M.A.O.: Human motion signatures: analysis, synthesis, recognition. In: *Proceedings International Conference on Pattern Recognition*, vol. 3, pp. 456–460, Quebec City (2002)
70. Vasilescu, M.A.O.: Incremental Multilinear SVD. In: *Proceedings Conference on ThRee-way methods In Chemistry And Psychology (TRICAP 06)* (2006)
71. Vasilescu, M.A.O.: A multilinear (Tensor) algebraic framework for computer graphics, Computer Vision, and Machine Learning, PhD dissertation, University of Toronto (2009)
72. Vasilescu, M.A.O.: Multilinear projection for face recognition via canonical decomposition. In: *Proceedings IEEE International Conference on Automatic Face Gesture Recognition (FG 2011)*, pp. 476–483 (2011)
73. Vasilescu, M.A.O., Kim, E.: Compositional hierarchical tensor factorization: representing hierarchical intrinsic and extrinsic causal factors. In: *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2019): Tensor Methods for Emerging Data Science Challenges Workshop* (2019)
74. Vasilescu, M.A.O., Kim, E., Zeng, X.S.: CausalX: causal explanations and block multilinear factor analysis. In: *2020 25th International Conference of Pattern Recognition (ICPR 2020)*, pp. 10736–10743 (2021)
75. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis for facial image recognition. In: *Proceedings International Conference on Pattern Recognition*, vol. 2, pp. 511–514, Quebec City (2002)
76. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: tensorFaces. In: *Proceedings European Conference on Computer Vision (ECCV 2002)*, pp. 447–460, Copenhagen, Denmark (2002)
77. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear subspace analysis of image ensembles. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, pp. 93–99, Madison, WI (2003)
78. Vasilescu, M.A.O., Terzopoulos, D.: TensorTextures: multilinear image-based rendering. *ACM Trans. Graphics* **23**(3), 336–342 (2004). *Proceedings ACM SIGGRAPH 2004 Conference*, Los Angeles, CA
79. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear independent components analysis. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 547–553, v.I, San Diego, CA (2005)
80. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear projection for appearance-based recognition in the tensor framework. In: *Proceedings 11th IEEE International Conference on Computer Vision (ICCV’07)*, pp. 1–8 (2007)
81. Vendrow, J., Haddock, J., Needell, D.: A generalized hierarchical nonnegative tensor decomposition (2021)
82. Vert, J.-P., Tsuda, K., Schölkopf, B.: A primer on kernel methods. *Kernel Meth. Comput. Biol.* **47**, 35–70 (2004)
83. Vlasic, D., Brand, M., Pfister, H., Popovic, J.: Face transfer with multilinear models. *ACM Trans. Graphics (TOG)* **24**(3), 426–433 (2005)
84. Wang, H., Ahuja, N.: A tensor approximation approach to dimensionality reduction. *Inter. J. Comput. Vision* **6**(3), 217–29 (2008)
85. Wang, W.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578 (2021)



86. Woodward, J.: *Making Things Happen: A Theory of Causal Explanation*, Oxford university press (2005)
87. Xiong, C., Liu, L., Zhao, X., Yan, S., Kim, T.K.: Convolutional fusion network for face verification in the wild. *IEEE Trans. Circuits Syst. Video Technol.* **26**(3), 517–528 (2016)
88. Yang, J., Gao, X., Zhang, D., Yang, J.: Kernel ICA: an alternative formulation and its application to face recognition. *Pattern Recogn.* **38**(10), 1784–87 (2005)
89. Ye, J.: Generalized low rank approximations of matrices. *Mach. Learn.* **61**(1), 167–191 (2005)



# Non-symmetrical Confidence Interval of AUC Measure Based on Cross-Validation

Yu Wang<sup>1,3</sup>(✉), Xiaoyan Zhao<sup>2</sup>, and Xingli Yang<sup>2,3</sup>

<sup>1</sup> School of Modern Educational Technology, Shanxi University,  
Taiyuan 030006, China

<sup>2</sup> School of Mathematics and Statistics, Shanxi University, Taiyuan 030006, China

<sup>3</sup> Key Laboratory of Complex Systems and Data Science (Shanxi University),  
Ministry of Education, Taiyuan 030006, China  
wangyu@sxu.edu.cn

**Abstract.** In pattern recognition research, model (algorithm) performance measure is a very important research direction, because the performance measure index has been used to evaluate the model performance throughout the whole process of model estimation, evaluation and selection. Currently, AUC (Area under the ROC (Receiver Operating Characteristic) Curve) measure has become a benchmark performance measure index for classification algorithm. In practical applications, the confidence interval technique of AUC measure is always used to measure the performance of classification algorithm. As we confirmed through simulated experiments, however, those widely used symmetrical confidence intervals with the form of Mean  $\pm$  SD (Standard Deviation) based on normal distribution assumption may be inappropriate and often exhibit low accuracy, this is because the distribution of AUC measure is actually non-symmetrical. Thus, a new non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validation is presented by theoretically analyzing its approximate distribution in this paper. Extensive simulated and real data experiments show that the proposed non-symmetrical confidence interval has higher degrees of confidence and shorter interval lengths than the benchmark symmetrical confidence intervals of AUC measure based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions.

**Keywords:** Confidence interval · AUC measure · Cross-validation · Degrees of confidence · Interval length

---

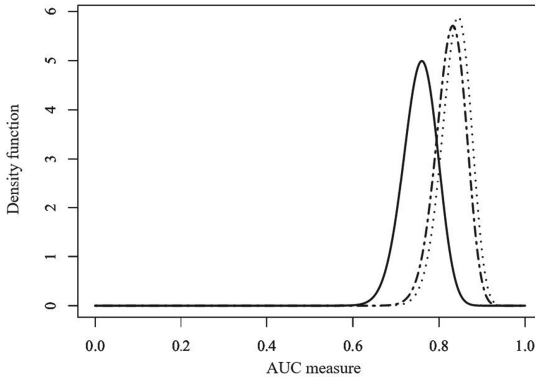
Supported by the National Natural Science Foundation of China under Grant 62076156, the Shanxi Scholarship Council of China under Grant 2023-013 and the Fundamental Research Program of Shanxi Province under Grants 202303021212023 and 202203021211305.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15309, pp. 439–454, 2025.  
[https://doi.org/10.1007/978-3-031-78189-6\\_28](https://doi.org/10.1007/978-3-031-78189-6_28)

# 1 Introduction

In the study of pattern recognition and machine learning, the classification problem such as diagnosis of disease and recognition of signal is one of their basic research contents [12]. The (pattern) classification or recognition is to establish a model (algorithm) or reasoning system describing the relationship between object features and categories and further to determine the categories of unknown samples based on the established model [17]. The classification problem generally includes two-class and multi-class classification problems. However, in practical applications, the multi-class classification problem can be often transformed into the two-class classification problem. Thus, this paper mainly focuses on the two-class classification problem.

In particular, the ROC (Receiver Operating Characteristic) curve and AUC (Area under the ROC Curve) measure have received widespread attention and widely applied in the fields of machine learning and artificial intelligence to better quantify the overall performance of algorithm in recent years [1, 9, 13, 18, 21, 25–27]. Therefore, this paper mainly focuses on the research of AUC measure.



**Fig. 1.** The density curves of AUC measure with different true positive rates ( $TPR$ ) and false positive rates ( $FPR$ )

Furthermore, in practical applications, in order to eliminate the influence of randomness, measure indexes based on  $K$ -fold cross-validation are often used to measure the performance of classification algorithm. However, the point estimation is trivial, without considering the estimation variance [15, 36]. For this reason, the widely used symmetrical confidence interval of AUC measure constructed by  $K$ -fold cross-validated  $t$  and corrected  $t$  distributions based on normal distribution assumption in the literature is proposed to measure the performance of classification algorithm. However, these symmetric confidence intervals often show low degree of confidence or long interval length, which can easily lead to liberal statistical inference results [14, 22].

Through theoretically analyzing the characteristic of AUC measure, we found that the distribution of AUC measure is actually non-symmetrical (the left tail of the AUC density curve is much longer than the right tail, and the peak value is skewed to the right), as shown in Fig. 1. At this time, it may be inappropriate to use a symmetrical distribution (such as  $t$  distribution) to approximate the distribution of AUC measure. Even more, it may lead to erroneous result or large deviation, because the value range of AUC measure is in the  $(0, 1)$  interval, however, the symmetrical confidence interval based on  $t$  distribution may exceed the range of  $(0, 1)$ , which is also verified by the subsequent experimental results.

Therefore, in order to effectively measure the performance of algorithm, it is very important to construct a faithful confidence interval of AUC measure with high degree of confidence and short interval length. The degree of confidence of a confidence interval refers to the probability that the confidence interval contains a true value, and interval length is used to measure the accuracy of the confidence interval. In view of this, for the two-class classification problem, this paper construct a new non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validation.

The contributions of this paper are as follows:

(1) By theoretically analyzing the characteristic of the distribution of the AUC measure, a non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated Beta distribution is proposed.

(2) Extensive simulated and real data experiments demonstrate that the proposed non-symmetrical confidence interval has higher degree of confidence and shorter interval length, which can effectively improve the performance of traditional symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the ROC curve, the AUC measure, and the macro-averaged and micro-averaged AUC measures based on  $K$ -fold cross-validation. Section 4 gives a detailed description of the non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated Beta distribution proposed in this paper. Section 5 compares the experimental performances of the confidence interval proposed in this paper with traditional confidence intervals through a large number of simulated and real data experiments to verify the superiority of the proposed confidence interval. Finally, we give the conclusion of this paper.

## 2 Related Work

The performance measure index widely used for classification model evaluation generally includes three categories. A brief overview is given as follows.

**Traditional Performance Measure Index.** Traditionally, the commonly used performance measure indexes for classification algorithm always include accuracy, error rate, precision, recall,  $F_1$  measure, sensitivity, specificity, true positive rate, false positive rate, and so on [15, 19, 24, 32, 33, 35, 36]. In particular, [8, 23, 30, 31] pointed out that although these measure indexes are proposed based

on different research backgrounds, they are all susceptible to the impact of category imbalance and cost sensitivity, so that they may not well reflect the real performance of classification algorithm in these situations.

**ROC Curve and AUC Measure.** In view of the problem of imbalanced data, threshold selection and multi-class classification faced in the measurement of classification performance, [13] proposed a ROC curve measure that is not sensitive to classification changes, which is a two-dimensional graph drawn with the true positive rate on the ordinate and the false positive rate on the abscissa. The closer the ROC curve is to the upper left corner of the graph, the better the performance of the corresponding classification algorithm, but when two ROC curves cross, it is difficult to identify which classification algorithm has the better performance. In view of this problem, the AUC measure based on the area under the ROC curve was proposed. Once it was proposed, this measure has received widespread attention and widely applied in the fields of machine learning and artificial intelligence because it can better quantify the overall performance of algorithm [1, 9, 18, 21, 25–27].

**Measure Index Based on Cross-Validation.** Furthermore, in practical applications, in order to eliminate the influence of randomness, measure indexes based on cross-validation are often used to measure the performance of classification algorithm. For example, for AUC measure,  $K$ -fold cross-validation divides the data into mutually exclusive and approximately equal  $K$  subsets, using  $K - 1$  subsets for training, and the remaining subset for testing. In this way,  $K$  confusion matrices can be obtained and the averaged AUC measures can be calculated based on these confusion matrices and used to measure model performance. However, the point estimation is trivial, without considering the estimation variance [7, 15, 36, 38, 39]. For this reason, the widely used symmetrical confidence interval of AUC measure constructed by  $K$ -fold cross-validated  $t$  and corrected  $t$  distributions based on normal distribution assumption in the literature [2, 3, 16] is proposed to measure the performance of classification algorithm. However, these symmetrical confidence intervals often show low degree of confidence or long interval length, which can easily lead to liberal statistical inference results. Thus, a non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated Beta distribution is proposed to more accurately evaluate model performance.

### 3 AUC Measures Based on $K$ -Fold Cross-Validation

In this section, we first introduce the definitions of ROC curve and AUC measure, and give the exact expressions of AUC measure with multiple thresholds. Then we present the averaged AUC measures based on  $K$ -fold cross-validation with macro-averaged and micro-averaged operators.

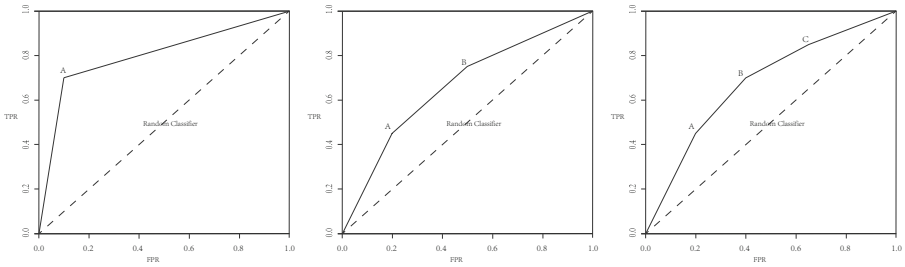
#### 3.1 ROC Curve and AUC Measure

For a specific two-class classification problem, the experimental results can be summarized in a  $2 \times 2$  confusion matrix consisting of true positive ( $TP$ ), false

positive ( $FP$ ), true negative ( $TN$ ), and false negative ( $FN$ ). Then, the true positive rate ( $TPR$ ) and false positive rate ( $FPR$ ) can be defined as follows:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{1}$$

The ROC curve is a two-dimensional graph plotted with  $TPR$  as vertical coordinate and  $FPR$  as horizontal coordinate. By using different thresholds for classification discrimination, type I and type II errors can be continuously changed, such that the change of ROC curve can clearly reflect the change of two type errors with the change of thresholds, which can effectively solve the problem of category imbalance and the performance of algorithm due to different costs of misclassification. The closer the ROC curve is to the upper left corner of graph, the better performance of the corresponding classification algorithm [5, 10, 11]. However, when two ROC curves cross, it is difficult to identify which classification algorithm has better performance. In view of this problem, the AUC measure based on the area under the ROC curve is proposed.



**Fig. 2.** The changes of the ROC curves with one, two, and three thresholds as the changes of  $TPR$  and  $FPR$  for two-class classification problem, where A, B, and C are thresholds

In the two-class classification problem, when only a single threshold of 0.5 is considered, the ROC curve has the shape shown in Fig. 2 and the AUC measure has the following expression [29, 37]:

$$AUC^{v=1} = \frac{1 + TPR_1 - FPR_1}{2} \tag{2}$$

Similarly, for the cases of two and three thresholds, the corresponding AUC measures have the following forms:

$$AUC^{v=2} = \frac{1}{2}TPR_1FPR_2 - \frac{1}{2}TPR_2FPR_1 + \frac{1}{2}TPR_2 - \frac{1}{2}FPR_2 + \frac{1}{2} \tag{3}$$

$$AUC^{v=3} = \frac{1}{2}TPR_1FPR_2 - \frac{1}{2}TPR_2FPR_1 + \frac{1}{2}TPR_2FPR_3 - \frac{1}{2}TPR_3FPR_2 + \frac{1}{2}TPR_3 - \frac{1}{2}FPR_3 + \frac{1}{2} \tag{4}$$

And analogously, when we take  $m$  different thresholds (the larger the  $m$ , the smoother its corresponding ROC curve), the form of AUC measure is as follows:

$$\begin{aligned}
 AUC^{v=m} &= \frac{1}{2} \sum_{i=1}^{m-1} (TPR_i FPR_{i+1} - TPR_{i+1} FPR_i) \\
 &+ \frac{1}{2} TPR_m - \frac{1}{2} FPR_m + \frac{1}{2}
 \end{aligned} \tag{5}$$

where,  $m$  is the number of thresholds,  $TPR_i$  and  $FPR_i$  refer to the coordinate values of  $TPR$  and  $FPR$  corresponding to the  $i$ -th ( $i = 1, \dots, m$ ) threshold, respectively.

### 3.2 Micro-averaged AUC Measure Based on $K$ -Fold Cross-Validation

In practice, in order to eliminate the effect by randomness, the  $K$ -fold cross-validation technique with multiple repetitions of training and testing is often used. Formally, the data set  $S$  is divided into  $K$  subsets with approximately same size and mutually exclusive, denoted as  $T_k, k = 1, 2, \dots, K$ . Let  $S_k$  denote the  $k$ -th training set obtained by removing the elements in  $T_k$  from the data set  $S$ . Thus,  $K$  training sets and  $K$  corresponding test sets are obtained. For each threshold, the averaged TP, FP, FN, and TN have the following form:

$$\begin{aligned}
 \overline{TP}_i(KCV) &= \frac{\sum_{k=1}^K TP_k^i}{K}, \quad \overline{FP}_i(KCV) = \frac{\sum_{k=1}^K FP_k^i}{K}, \\
 \overline{FN}_i(KCV) &= \frac{\sum_{k=1}^K FN_k^i}{K}, \quad \overline{TN}_i(KCV) = \frac{\sum_{k=1}^K TN_k^i}{K}.
 \end{aligned}$$

Then, the micro-averaged AUC measure based on  $K$ -fold cross-validation with multiple thresholds can be obtained:

$$\begin{aligned}
 AUC_{KCV}^{mic, v=m} &= \frac{1}{2} \sum_{i=1}^{m-1} \left( TPR_{i(KCV)}^{mic} FPR_{i+1(KCV)}^{mic} - TPR_{i+1(KCV)}^{mic} FPR_i^{mic} \right) \\
 &+ \frac{1}{2} TPR_m^{mic}(KCV) - \frac{1}{2} FPR_m^{mic}(KCV) + \frac{1}{2}.
 \end{aligned} \tag{6}$$

where

$$TPR_{i(KCV)}^{mic} = \frac{\overline{TP}_i(KCV)}{\overline{TP}_i(KCV) + \overline{FN}_i(KCV)}, \quad FPR_{i(KCV)}^{mic} = \frac{\overline{FP}_i(KCV)}{\overline{FP}_i(KCV) + \overline{TN}_i(KCV)}.$$

### 3.3 Macro-averaged AUC Measure Based on $K$ -Fold Cross-Validation

The macro-averaged AUC measure based on  $K$ -fold cross-validation is the average of  $K$  AUC measures computed based on  $K$  training and test sets. Specifically,

if denoting the  $AUC_k, k = 1, 2, \dots, K$  be the AUC measure computed from the  $k$ th training and test sets, the result of averaging these  $K$  AUC measure values is:

$$AUC_{KCV}^{mac,v=m} = \frac{\sum_{k=1}^K AUC_k^{v=m}}{K}. \tag{7}$$

[37] had proved that the macro-averaged and the micro-averaged AUC measures with single threshold based on  $K$ -fold cross-validation are identical, that is,  $AUC_{KCV}^{mac,v=1} = AUC_{KCV}^{mic,v=1}$ . However, when AUC measure is defined by multiple thresholds, the macro-averaged and the micro-averaged AUC measures based on  $K$ -fold cross-validation have a more complex form and thus it is hard to prove that they are identical. Even so, we have experimentally verified that their differences are actually very small in a variety of experimental situations. Thus, in this paper, we only provide the confidence interval of macro-averaged AUC measure based on  $K$ -fold cross-validation.

### 4 Confidence Interval of AUC Measure Based on $K$ -Fold Cross-Validated Beta Distribution

In this section, we present three confidence interval techniques for AUC measure based on  $K$ -fold cross-validation. The first two are symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions commonly used in the literature, and the third is a non-symmetrical confidence interval based on  $K$ -fold cross-validated Beta distribution proposed in this paper.

#### 4.1 Confidence Interval of AUC Measure Based on $K$ -Fold Cross-Validated $t$ -Distribution

Symmetrical confidence intervals based on  $t$ -distribution are widely used in the machine learning research [4, 20, 28, 40]. In general, symmetrical confidence intervals with a confidence level of  $1 - \alpha$  have the following form:

$$\left[ \hat{\mu} - c\sqrt{\hat{\sigma}^2}, \hat{\mu} + c\sqrt{\hat{\sigma}^2} \right], \tag{8}$$

where,  $\hat{\mu}$  is the sample mean,  $\hat{\sigma}^2$  is the sample variance, and  $c$  is the percentile of  $t$  distribution. Thus, the symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated  $t$  distribution can be written as:

$$CI_{AUC_{t(KCV)}^{v=m}} = \left[ AUC_{KCV}^{mac,v=m} - c_{K-1,1-\frac{\alpha}{2}} \sqrt{\hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2}, \right. \\ \left. AUC_{KCV}^{mac,v=m} + c_{K-1,1-\frac{\alpha}{2}} \sqrt{\hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2} \right], \tag{9}$$

where

$$\hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2 = \frac{1}{K(K-1)} \sum_{k=1}^K (AUC_k^{v=m} - AUC_{KCV}^{mac,v=m})^2.$$



### 4.2 Confidence Interval of AUC Measure Based on Corrected $K$ -Fold Cross-Validated $t$ -Distribution

[3] pointed out that the correlation between different folds in  $K$ -fold cross-validation cannot be ignored when calculating their variance, otherwise the variance will be grossly underestimated. For this reason, [16] proposed a corrected  $K$ -fold cross-validated  $t$  test based on corrected  $K$ -fold cross-validated variance. That is, if we let  $\hat{\mu} = AUC_{KCV}^{mac,v=m}$  and  $\hat{\sigma}^2 = \hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2 / (1 - \rho_{AUC_{KCV}^{mac,v=m}})$ , the confidence interval of AUC measure based on corrected  $K$ -fold cross-validated  $t$  distribution is:

$$CI_{AUC_{Ct(KCV)}^{v=m}} = \left[ AUC_{KCV}^{mac,v=m} - c_{K-1,1-\frac{\alpha}{2}} \sqrt{\frac{\hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2}{1 - \rho_{AUC_{KCV}^{mac,v=m}}}}, \right. \\ \left. AUC_{KCV}^{mac,v=m} + c_{K-1,1-\frac{\alpha}{2}} \sqrt{\frac{\hat{\sigma}_{AUC_{KCV}^{mac,v=m}}^2}{1 - \rho_{AUC_{KCV}^{mac,v=m}}}} \right], \quad (10)$$

where,  $\rho_{AUC_{KCV}^{mac,v=m}}$  is the correlation coefficient and [16] recommended a conservative empirical estimation of  $\hat{\rho}_{AUC_{KCV}^{mac,v=m}} = 0.7$ .

### 4.3 Confidence Interval of AUC Measure Based on $K$ -Fold Cross-Validated Beta Distribution

For the convenience of the subsequent theoretical derivation, we first introduce some lemmas.

**Lemma 1.** [15] *In a two-class classification problem, let  $TPR = \frac{TP}{TP+FN}$ , and  $FPR = \frac{FP}{FP+TN}$  be the true positive rate and the false positive rate, then the posterior distributions of  $TPR$ , and  $FPR$  are  $TPR|D \sim Be(TP + \lambda, FN + \lambda)$ , and  $FPR|D \sim Be(FP + \lambda, TN + \lambda)$  respectively, where  $D = (TP, FP, FN, TN)$ ,  $TPR \sim Be(\lambda, \lambda)$ ,  $FPR \sim Be(\lambda, \lambda)$ ,  $\lambda$  is the prior parameter.*

**Lemma 2.** [6, 34] *The product and sum of two mutually independent random variables that both follow Beta distribution still approximately follow Beta distribution, i.e., if random variables  $X \sim Be(a_1, b_1)$  and  $Y \sim Be(a_2, b_2)$ , the random variables  $Z = X \cdot Y$  and  $W = X + Y$  also approximately follow Beta distribution.*

Recalling that the expression of AUC measure is shown in Eq. (5), we know that it is mainly composed of four parts:  $TPR_i FPR_{i+1}$ ,  $TPR_{i+1} FPR_i$ ,  $TPR_m$  and  $FPR_m$ . From Lemmas 1 and 2, we know that these four parts all approximately follow Beta distribution, then we can deduce that  $AUC^{v=m}$  also approximately follows Beta distribution. Furthermore, for the macro-averaged AUC measure based on  $K$ -fold cross-validation with the mean of  $K$   $AUC^{v=m}$ s, its distribution should be also close to a beta distribution.

**Theorem 1.** *If assuming that  $TPR_i, FPR_{i+1}, TPR_{i+1}, FPR_i, FPR_m,$  and  $TPR_m$  ( $i = 1, \dots, m - 1$ ) are independent, the macro-averaged AUC measure based on  $K$ -fold cross-validation with multiple thresholds approximately follows a Beta distribution, that is*

$$AUC_{KCV}^{mac,v=m} \approx Be(a_{KCV}^{mac,v=m}, b_{KCV}^{mac,v=m}), \tag{11}$$

where

$$\begin{aligned} a_{KCV}^{mac,v=m} &= \frac{E}{V} (E - E^2 - V), \\ b_{KCV}^{mac,v=m} &= \frac{1 - E}{V} (E - E^2 - V), \\ E &= \frac{\sum_{k=1}^K E(AUC_k^{v=m} | D_k)}{K}, \\ V &= \frac{\sum_{k=1}^K Var(AUC_k^{v=m} | D_k)}{K^2}. \end{aligned}$$

*Proof.* By equating the first and second moments of  $AUC_{KCV}^{mac,v=m}$  and the random variable following beta distribution, we have

$$E(AUC_{KCV}^{mac,v=m}) = \frac{a_{KCV}^{mac,v=m}}{a_{KCV}^{mac,v=m} + b_{KCV}^{mac,v=m}},$$

$$Var(AUC_{KCV}^{mac,v=m}) = \frac{a_{KCV}^{mac,v=m} b_{KCV}^{mac,v=m}}{(a_{KCV}^{mac,v=m} + b_{KCV}^{mac,v=m})^2 (a_{KCV}^{mac,v=m} + b_{KCV}^{mac,v=m} + 1)}.$$

From this, one can show that

$$\begin{aligned} a_{KCV}^{mac,v=m} &= \frac{E(AUC_{KCV}^{mac,v=m})}{Var(AUC_{KCV}^{mac,v=m})} [E(AUC_{KCV}^{mac,v=m}) \\ &\quad - E(AUC_{KCV}^{mac,v=m})^2 - Var(AUC_{KCV}^{mac,v=m})], \end{aligned}$$

$$\begin{aligned} b_{KCV}^{mac,v=m} &= \frac{1 - E(AUC_{KCV}^{mac,v=m})}{Var(AUC_{KCV}^{mac,v=m})} [E(AUC_{KCV}^{mac,v=m}) \\ &\quad - E(AUC_{KCV}^{mac,v=m})^2 - Var(AUC_{KCV}^{mac,v=m})]. \end{aligned}$$

From Theorem 1 we can easily obtain the non-symmetrical confidence interval for the AUC measure based on  $K$ -fold cross-validated Beta distribution with multiple thresholds:

$$CI_{AUC_{Beta(KCV)}^{v=m}} = \left[ Be(a_{KCV}^{mac,v=m}, b_{KCV}^{mac,v=m})_{\frac{\alpha}{2}}, Be(a_{KCV}^{mac,v=m}, b_{KCV}^{mac,v=m})_{1-\frac{\alpha}{2}} \right]. \tag{12}$$

where  $Be(\cdot, \cdot)_{\frac{\alpha}{2}}$  is the percentile of the Beta distribution.

## 5 Experimental Results and Analysis

In this section, simulated and real data experiments are conducted to compare the degrees of confidence and interval lengths of the symmetrical confidence intervals of AUC measure based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions, and the approximately non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated Beta distribution under multiple classifiers of classification tree (CT), support vector machines with Gaussian kernel function (SVM), and naive Bayes (NB). In the experiments, we choose widely used  $K = 10$ ,  $K = 5$  and  $K = 2$  for  $K$ -fold cross-validation. The prior parameter  $\lambda$  in the Beta distribution is taken as 1, and the confidence level  $1 - \alpha = 0.95$ , i.e.,  $\alpha = 0.05$ . All experiments were repeated 1,000 times to take into account the effect of randomness in the training and test sets.

### 5.1 Experimental Settings

**Simulated Data:** Consider a two-class classification problem with data set  $Z = (X, Y)$ , where  $X = (x_1, x_2, \dots, x_d)$  is the  $d$ -dimensional feature vector,  $Y = \{0, 1\}$  is the binary response variable,  $P(Y = 1) = P(Y = 0) = 0.5$ ,  $X|Y = 0 \sim N(\mu_0, \Sigma_0)$ , and  $X|Y = 1 \sim N(\mu_1, \Sigma_1)$ . Here, we take  $\mu_0 = 0_d$ ,  $\Sigma_0 = I_d$ ,  $\mu_1 = \beta_1 1_d$ ,  $\Sigma_1 = \beta_2 \Sigma_0$ , where  $0_d$  and  $1_d$  refer to the  $d$ -dimensional vectors with all elements 0 and 1, and  $I_d$  refers to the  $d$ -order identity matrix. The total sample size  $n$  is 200 and 1,000.

**Real Data:** The MAGIC Gamma Telescope data set from the UCI database is used to simulate the localization of high-energy  $\gamma$  particles in the atmospheric cherenkov telescope. Specifically, it contains 10 feature variables (long axis, short axis, sum, continuous ratio, etc.) and 19,020 samples. The category labels are  $g$  class (0) and  $h$  class (1), where  $g$  class represents the signal with 12,332 samples, and  $h$  class represents the background with 6,688 samples.

A data set for identifying the letters of the roman alphabet comprises 20,000 examples described by 16 features (pixel position on the left side of the rectangle (horizontal position), pixel position on the bottom side of the rectangle (vertical position), rectangle width, rectangle height, etc.). The 26 letters represent 26 categories, and in this experiment we turn it into a two-class (A-M versus N-Z) classification problem. We sample, with replacement, 200 (1,000) examples from the 19,020 (20,000) examples available in the Telescope (Letter) data set.

### 5.2 Experimental Results and Analysis of Simulated Data

Table 1 presents the results of degrees of confidence and interval lengths of three confidence intervals of AUC measure based on  $K$ -fold cross-validation ( $K=2, 5, 10$ ) under the naive Bayes, classification tree, and support vector machine classifiers. First, from the table we can see that the degrees of confidence of confidence interval of AUC measure based on  $K$ -fold cross-validated  $t$  distribution

under all three classifiers are all below 96.00%, with a maximum of 95.30%. By correcting the variance of  $t$  statistic, the confidence interval of AUC measure based on corrected  $K$ -fold cross-validated  $t$  distribution has a better degree of confidence. In most cases, it can reach 96.00%, even 99.90%. For example, in the case of  $n = 200, d = 5, \beta = (0.2, 3)$ , the confidence interval of AUC measure based on corrected 10-fold cross-validated  $t$  distribution has a degree of confidence of 99.80%. However, for the non-symmetrical confidence interval proposed in this paper, the degrees of confidence are all close to 100.00% in all cases.

**Table 1.** Degrees of Confidence and interval lengths of three confidence intervals on the simulated data for the cases of  $n = 200, d = 5, \beta = (1, 2)$  (Case 1),  $n = 1000, d = 5, \beta = (1, 2)$  (Case 2),  $n = 200, d = 5, \beta = (0.2, 3)$  (Case 3),  $n = 1000, d = 5, \beta = (0.2, 3)$  (Case 4),  $n = 200, d = 5, \beta = (0.2, 3)$  (Case 5) and  $n = 1000, d = 5, \beta = (0.2, 3)$  (Case 6), where CT, NB, and SVM refer to classification tree, naive Bayes, and support vector machine classifiers respectively,  $d$  is the feature dimension, and  $n$  is the total sample size

		Simulated Data					
		CT		NB		SVM	
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
$CI_{AUC_{t(2CV)}^{v=m}}$	DOC	93.10%	94.50%	94.30%	95.10%	92.50%	93.50%
	IL	0.740	0.330	0.491	0.219	0.428	0.215
$CI_{AUC_{t(5CV)}^{v=m}}$	DOC	94.00%	91.50%	93.50%	95.30%	91.70%	93.00%
	IL	0.168	0.073	0.130	0.057	0.134	0.058
$CI_{AUC_{t(10CV)}^{v=m}}$	DOC	91.20%	89.60%	93.30%	94.50%	91.80%	92.50%
	IL	0.137	0.059	0.114	0.048	0.120	0.050
$CI_{AUC_{Ct(2CV)}^{v=m}}$	DOC	96.40%	96.90%	96.80%	97.00%	95.90%	96.10%
	IL	1.351	0.602	0.896	0.399	0.782	0.392
$CI_{AUC_{Ct(5CV)}^{v=m}}$	DOC	99.10%	98.60%	98.90%	99.40%	97.90%	99.00%
	IL	0.307	0.134	0.237	0.103	0.244	0.105
$CI_{AUC_{Ct(10CV)}^{v=m}}$	DOC	98.90%	99.30%	99.80%	99.60%	99.40%	99.60%
	IL	0.251	0.107	0.209	0.088	0.219	0.091
$CI_{AUC_{Beta(2CV)}^{v=m}}$	DOC	100.00%	100.00%	99.80%	100.00%	100.00%	100.00%
	IL	0.147	0.066	0.102	0.043	0.111	0.046
$CI_{AUC_{Beta(5CV)}^{v=m}}$	DOC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	IL	0.128	0.057	0.100	0.043	0.106	0.045
$CI_{AUC_{Beta(10CV)}^{v=m}}$	DOC	99.90%	100.00%	100.00%	100.00%	100.00%	100.00%
	IL	0.124	0.054	0.101	0.043	0.107	0.045

Second, for interval length, the confidence intervals of AUC measure based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions all

exhibit long interval lengths. The confidence interval technique of AUC measure based on corrected  $K$ -fold cross-validated  $t$  distribution improves the degree of confidence by correcting the degree of freedom of variance estimation, however, the cost is that the interval length of the confidence interval of AUC measure based on corrected  $K$ -fold cross-validated  $t$  distribution increases by about two times compared with the confidence interval of AUC measure based on  $K$ -fold cross-validated  $t$  distribution. For example, for the SVM classifier, the interval lengths of confidence intervals of AUC measure based on 2, 5, and 10-fold cross-validated  $t$  distributions are 0.428, 0.215, 0.134, 0.058, 0.120, and 0.050, respectively, however, they are 0.782, 0.392, 0.244, 0.105, 0.219, and 0.091 for the confidence intervals of AUC measure based on corrected 2, 5, and 10-fold cross-validated  $t$  distributions. The latter is nearly twice as many as the former.

However, there is no such problem with the proposed approximately non-symmetrical confidence interval. The proposed confidence interval technique has shorter interval length than these two symmetrical confidence interval while maintaining high degree of confidence. For example, from Table 1 we can see that in the case of  $n = 200, d = 5, \beta = (1, 2)$ , the interval lengths of confidence intervals of AUC measure based on 2-fold cross-validated  $t$  and corrected 2-fold cross-validated  $t$  distributions are 0.740 and 1.351, respectively, however, the proposed confidence interval is only 0.147 while maintaining the degree of confidence be 100.00%.

Overall, whether for degree of confidence or for interval length, the proposed non-symmetrical confidence interval based on  $K$ -fold cross-validated Beta distribution is superior to the other two symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions. However, in practical applications, a fundamental principle for selecting the confidence interval is to select the one with the shortest interval length for an acceptable degree of confidence. With the adopted degree of confidence of 95.00%, the interval lengths of the proposed non-symmetrical confidence interval are only half that of the symmetrical confidence interval based on corrected  $K$ -fold cross-validated  $t$  distribution, and sometimes shorter. That is, the proposed non-symmetrical confidence interval technique based on  $K$ -fold cross-validated Beta distribution is significantly better than the symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions.

Besides, with different folds of 2, 5, and 10 in the  $K$ -fold cross-validation, these confidence intervals behave differently. The performances of the confidence intervals based on 5 and 10-fold cross-validations are all better than that of the confidence intervals based on 2-fold cross-validation. When the sample size increases from 200 to 1000, there was little change to the degree of confidence for all confidence intervals. However, their interval lengths decrease by approximately half or two thirds.

**Table 2.** Degrees of Confidence and interval lengths of three confidence intervals on Telescope and Letter data sets

		Telescope Data set						Letter Data set			
		CT		NB		SVM		CT		NB	
		$n = 200$	$n = 1000$	$n = 200$	$n = 1000$	$n = 200$	$n = 1000$	$n = 200$	$n = 1000$	$n = 200$	$n = 1000$
$CI_{AUC_{t(2CV)}^{m=}}$	DOC	93.70%	92.60%	96.80%	92.60%	91.60%	92.10%	91.80%	92.50%	91.40%	96.00%
	IL	0.709	0.336	0.981	0.336	0.560	0.229	0.792	0.420	0.612	0.350
$CI_{AUC_{t(5CV)}^{v=}}$	DOC	92.70%	93.50%	94.20%	95.20%	94.40%	95.80%	89.80%	94.50%	92.70%	96.80%
	IL	0.170	0.075	0.190	0.086	0.140	0.053	0.186	0.086	0.178	0.077
$CI_{AUC_{t(10CV)}^{m=}}$	DOC	88.80%	89.30%	94.50%	96.00%	95.10%	95.90%	90.00%	89.20%	93.30%	96.80%
	IL	0.140	0.060	0.157	0.068	0.117	0.044	0.155	0.066	0.149	0.062
$CI_{AUC_{Ct(2CV)}^{v=}}$	DOC	96.10%	96.60%	98.70%	96.60%	96.10%	95.50%	95.30%	96.20%	95.10%	97.90%
	IL	1.295	0.614	1.791	0.614	1.022	0.419	1.447	0.766	1.118	0.639
$CI_{AUC_{Ct(5CV)}^{m=}}$	DOC	99.30%	98.50%	99.20%	99.60%	98.70%	99.20%	98.50%	99.40%	99.10%	99.70%
	IL	0.311	0.136	0.347	0.157	0.256	0.097	0.340	0.157	0.325	0.141
$CI_{AUC_{Ct(10CV)}^{v=}}$	DOC	98.90%	99.30%	99.70%	100.00%	99.80%	99.50%	98.40%	98.90%	99.70%	100.00%
	IL	0.256	0.110	0.288	0.125	0.214	0.081	0.283	0.120	0.272	0.113
$CI_{AUC_{Beta(2CV)}^{v=}}$	DOC	100.00%	99.80%	100.00%	99.90%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	IL	0.150	0.068	0.168	0.068	0.117	0.043	0.171	0.082	0.151	0.062
$CI_{AUC_{Beta(5CV)}^{v=}}$	DOC	100.00%	100.00%	99.80%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	IL	0.132	0.057	0.144	0.063	0.106	0.039	0.148	0.066	0.135	0.056
$CI_{AUC_{Beta(10CV)}^{m=}}$	DOC	99.90%	100.00%	100.00%	100.00%	99.80%	100.00%	100.00%	100.00%	100.00%	100.00%
	IL	0.128	0.054	0.140	0.061	0.106	0.039	0.143	0.060	0.134	0.055

### 5.3 Experimental Results and Analysis of Real Data

Table 2 gives the experimental comparison results of three confidence intervals of AUC measure based on  $K$ -fold cross-validation for three classifiers on Telescope and Letter data sets. Similar to the simulated data situation, the symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  distribution exhibits a degraded degree of confidence. In 21 of 30 cases, the degrees of confidence fell below 95.00%, as shown in Table 2. By correcting the variance of  $t$  statistic, the symmetrical confidence interval based on corrected  $K$ -fold cross-validated  $t$  distribution elevates the degree of confidence of that based on  $K$ -fold cross-validated  $t$  distribution. They all exceed 95.00%. For the proposed non-symmetrical confidence interval based on  $K$ -fold cross-validated Beta distribution has approximately 100.00% degree of confidence in all cases.

Even though for the case of low degree of confidence of the symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  distribution, the proposed non-symmetrical confidence interval remains have shortest interval length. For example, in the case of  $n = 200$  and CT classifier on Telescope data set, the degree of confidence and interval length of the symmetrical confidence intervals based on 10-fold cross-validated  $t$  distribution are 88.80% and 0.140, respectively. However, they are 99.90% and 0.128 for the proposed non-symmetrical confidence interval.

With an acceptable degree of confidence (above 95.00%), interval lengths of the proposed non-symmetrical confidence interval are about half that of the

symmetrical confidence interval based on corrected  $K$ -fold cross-validated  $t$  distribution. And the results in Table 2 show that the interval length also decreases by half as the sample size changed from 200 to 1,000 for all three confidence intervals. This implies that the sample size has a significant impact on the interval length of confidence interval.

**Remark:** It is well known that the AUC measure value is between 0 and 1, however, the symmetrical confidence interval based on  $t$  distribution may exceed the range of (0, 1). For example, the simulated and real experiments in Tables 1 and 2 with  $n = 200$  show that the interval lengths of the confidence intervals of AUC measure based on corrected 2-fold cross-validated  $t$  distribution is 1.351, 1.295, 1.791, 1.022, 1.447, and 1.118, which obviously exceeds the limit value of 1. In this case, it may be inappropriate using the symmetrical confidence interval based on  $t$  distribution to measure the classification performance of algorithm.

## Conclusion

In this paper, we construct a non-symmetrical confidence interval of AUC measure based on  $K$ -fold cross-validated Beta distribution with multiple thresholds by theoretically analyzing its approximate posterior distribution. Extensive experimental results demonstrate that the proposed non-symmetrical confidence interval has higher degree of confidence and shorter interval length, which can effectively improve the performance of traditional symmetrical confidence intervals based on  $K$ -fold cross-validated  $t$  and corrected  $K$ -fold cross-validated  $t$  distributions. This also provides a new idea and direction for the future research of AUC measure.

To develop this view, we will further show how the proposed non-symmetrical confidence interval can be improved to make it suitable for more complex situations such as non-independence.

**Acknowledgements.** The experiments are supported by High Performance Computing System of Shanxi University. Yu Wang is the corresponding author.

## References

1. Atapattu, S., Tellambura, C., Jiang, H.: Analysis of area under the roc curve of energy detection. *IEEE Trans. Wireless Commun.* **9**(3), 1216–1225 (2010)
2. Bayle, P., Bayle, A., Janson, L., Mackey, L.W.: Cross-validation confidence intervals for test error. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 16339–16350 (2020)
3. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of  $k$ -fold cross-validation. *J. Mach. Learn. Res.* **5**, 1089–1105 (2004)
4. Bisani, M., Ney, H.: Bootstrap estimates for confidence intervals in ASR performance evaluation. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 216–219 (2004)
5. Buja, A., Stuetzle, W., Shen, Y.: *Loss functions for binary class probability estimation and classification: Structure and applications*, University of Pennsylvania (2005)

6. Chen, W., Mi, S., Wei, Z.: A method of determining what distribution the product (or sum) of two random variables satisfies. *Appl. Stat. Manage.* **22**, 23–27 (2003)
7. Coleman, T., Peng, W., Mentch, L.: Scalable and efficient hypothesis testing with random forests. *J. Inf. Eng. Univ.* **23**(170), 1–35 (2022)
8. Cortes, C., Mohri, M.: Auc optimization vs. error rate minimization. *Adv. Neural Inf. Proce. Syst.* 313–320 (2003)
9. Cortes, C., Mohri, M.: Confidence intervals for the area under the roc curve. *Adv. Neural Inf. Proce. Syst.* 305–312 (2004)
10. Costa, E.P., Lorena, A.C., Carvalho, A.C., Freitas, A.A.: A review of performance evaluation measures for hierarchical classifiers. In: *Association for the Advance of Artificial Intelligence*, pp. 1–6 (2007)
11. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**(1), 1–30 (2006)
12. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern classification*, 2nd edition. Wiley (2001)
13. Fawcett, T.: An introduction to roc analysis. *Pattern Recogn. Lett.* **27**(8), 861–874 (2006)
14. Feng, Y., Tang, Z., Liu, Q.: Non-asymptotic confidence intervals of off-policy evaluation: Primal and dual bounds. In: *Proceedings of the 9th International Conference on Learning Representations* (2021)
15. Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: *European conference on information retrieval*, pp. 345–359. Springer (2005)
16. Grandvalet, Y., Bengio, Y.: *Hypothesis Testing for Cross-validation*. Montreal Université de Montreal, Operationnelle DdLeR (2006)
17. Hastie, T., Tibshirani, R.J., Friedman, J.: *The elements of statistical learning: Data mining, inference, and prediction*. *Math. Intell.* **27**(2), 83–85 (2005)
18. Kamalov, F., Leung, H.H.: Roc curve model under pareto distribution. *Appl. Math. Sci.* **10**, 461–466 (2016)
19. Kazan, Z., Shi, K., Groce, A., Bray, A.: The test of tests: a framework for differentially private hypothesis testing. In: *Proceedings of the 40th International Conference on Machine Learning*, pp. 16131–16151 (2023)
20. Keller, M., Bengio, S., Wong, S.Y.: Benchmarking non-parametric statistical tests. *Adv. Neural Inf. Proce. Syst.* 651–658 (2005)
21. LeDell, E., Petersen, M.L., van der Laan, M.J.: Computationally efficient confidence intervals for cross-validated area under the roc curve estimates. *Electron. J. Stat.* **9**(1), 1583–1607 (2015)
22. Li, Z., Xie, C., Wang, Q.: Asymptotic normality and confidence intervals for prediction risk of the min-norm least squares estimator. In: *Proceedings of the 38th International Conference on Machine Learning*, pp. 6533–6542 (2021)
23. Ling, C.X., Huang, J., Zhang, H.: Auc: a statistically consistent and more discriminating measure than accuracy. In: *Proceedings of the 18th international joint conference on Artificial intelligence*, pp. 519–524 (2003)
24. Liu, Z., Li, Z., Wang, J., He, Y.: Full bayesian significance testing for neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8841–8849 (2024)
25. Lobo, J.M., Jiménez-Valverde, A., Real, R.: Auc: a misleading measure of the performance of predictive distribution models. *Glob. Ecol. Biogeogr.* **17**(2), 145–151 (2008)
26. Ma, H., Bandos, A.I., Gur, D.: On the use of partial area under the roc curve for comparison of two diagnostic tests. *Biom. J.* **57**(2), 304–320 (2015)



27. Marrocco, C., Duin, R.P.W., Tortorella, F.: Maximizing the area under the roc curve by pairwise feature combination. *Pattern Recogn.* **41**(6), 1961–1974 (2008)
28. Nadeau, C., Bengio, Y.: Inference for the generalization error. *Mach. Learn.* **52**, 239–281 (2003)
29. Pomenkova, J., Malach, T.: Comparing classifier’s performance based on confidence interval of the roc. *Radioengineering* **27**(3), 827–834 (2018)
30. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 43–48 (1997)
31. Provost, F.J., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453 (1998)
32. Ren, Y., Zhang, H., Xia, Y., Guan, J., Zhou, S.: Multi-level wavelet mapping correlation for statistical dependence measurement: Methodology and performance. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6499–6506 (2023)
33. Wang, C., Liu, Y., Wang, X., Yan, G.: Appraisal identification of classifier’s performance. *Electron. Des. Eng.* **19**(8), 13–15 (2011)
34. Wang, X., Wei, Z.: Application of numerical value calculation and computer simulation on two beta random variables’ product. *J. Inf. Eng. Univ.* **4**(1), 82–85 (2003)
35. Wang, Y., Li, J.: Credible intervals for precision and recall based on a k-fold cross-validated beta distribution. *Neural Comput.* **28**(8), 1694–1722 (2016)
36. Wang, Y., Li, J., Li, Y., Wang, R., Yang, X.: Confidence interval for  $f_1$  measure of algorithm performance based on blocked  $3 \times 2$  cross-validation. *IEEE Trans. Knowl. Data Eng.* **27**(3), 651–659 (2015)
37. Wang, Y., Zhao, X., Yang, X., Li, J.: Confidence interval of AUC measure based on k-fold cross-validated beta distribution. *J. Syst. Sci. Math. Sci.* **40**(9), 51–64 (2020)
38. Wooldridge, M.J., Dy, J.G., Natarajan, S.: Permutation-based hypothesis testing for neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 14306–14314 (2024)
39. Yang, W., Poyiadzi, R., Twomey, N.: Hypothesis testing for class-conditional noise using local maximum likelihood. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21744–21752 (2024)
40. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–49 (1999)



# Visualizing and Generalizing Integrated Attributions

Ethan Payne<sup>1</sup> , David Patrick<sup>1,2</sup> , and Amanda S. Fernandez<sup>1</sup>  

<sup>1</sup> The University of Texas at San Antonio, San Antonio, TX, USA  
amanda.fernandez@utsa.edu

<sup>2</sup> Texas State University, San Marcos, TX, USA

**Abstract.** Explainability and attribution for deep neural networks remains an open area of study due to the importance of adequately interpreting the behavior of such ubiquitous learning models. The method of expected gradients [10] reduced the baseline dependence of integrated gradients [27] and allowed for improved interpretability of attributions as representative of the broader gradient landscape, however both methods are visualized using an ambiguous transformation which obscures attribution information and neglects to distinguish between color channels. While expected gradients takes an expectation over the entire dataset, this is only one possible domain in which an explanation can be contextualized. In order to generalize the larger family of attribution methods containing integrated gradients and expected gradients, we instead frame each attribution as a volume integral over a set of interest within the input space, allowing for new levels of specificity and revealing novel sources of attribution information. Additionally, we demonstrate these new unique sources of feature attribution information using a refined visualization method which allows for both signed and unsigned attributions to be visually salient for each color channel. This new formulation provides a framework for developing and explaining a much broader family of attribution measures, and for computing attributions relevant to diverse contexts such as local and non-local neighborhoods. We evaluate our novel family of attribution measures and our improved visualization method using qualitative and quantitative approaches with the CIFAR10 and ImageNet datasets and the Quantus XAI library.

**Keywords:** Attribution · Saliency · Influence · Integrated Gradients · Expected Gradients · Explainability · Causal Inference · Visualization

## 1 Introduction

While gradient-based approaches to feature attribution for deep neural networks are both intuitive and relatively easy to implement, established methods such as

---

Supported by the National Science Foundation under Grant No. 2134237. Code: <https://github.com/UTSA-VAIL/Visualizing-and-Generalizing-Integrated-Attributions>.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025  
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15309, pp. 455–470, 2025.  
[https://doi.org/10.1007/978-3-031-78189-6\\_29](https://doi.org/10.1007/978-3-031-78189-6_29)

integrated gradients [27] which rely on paths to fixed external reference inputs often lack a compelling justification for why certain baselines should be chosen over others. There may be situations and applications which may support obvious baselines, but as noted by Erion et al. [10], this is often not the case. Many of the shortcomings of integrated gradients were alleviated by computing the expected gradients as a Monte Carlo integral over the training dataset, however this approach does not succeed in completely generalizing the original intuition of integrated gradients to a comprehensive family of attribution measures.

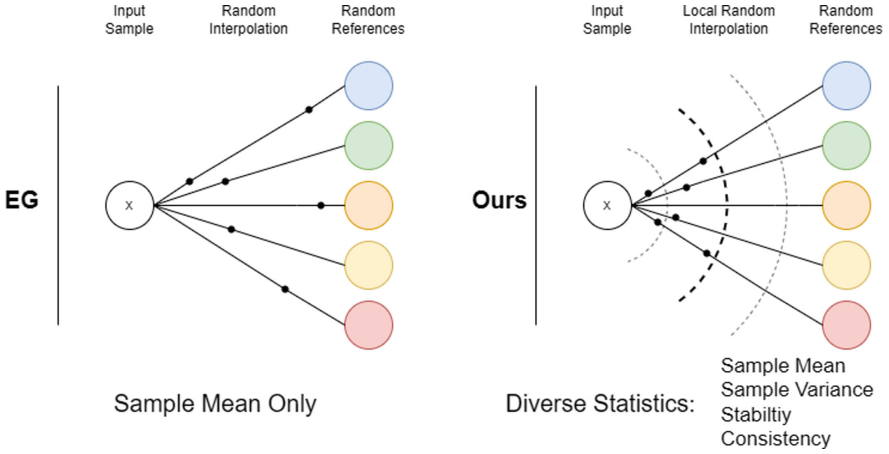
We present a generalization of integrated gradients [27] and expected gradients [10] which also encompasses a diverse family of other attribution measures. By formulating the expected gradients in terms of a volume integral rather than a path integral, we obtain an attribution method which is immediately generalizable to any deep learning application, and which can be easily iterated upon. We note that our formulation has similar implementation requirements as expected gradients while allowing us to access several unique sources of attribution information which were previously not utilized. Using our new formulation of *generalized integrated gradients*, we are able to identify distinct paradigms of attribution information corresponding to input locality.

Additionally, leverage our new formulation to develop three new measures of gradient variance, stability, and consistency, which each quantify a unique aspect of model behavior. Gradient variance quantifies the dispersion of model gradients, and results in attributions which provide improved visual salience over expected gradients. Our stability and consistency measures incorporate angular information to characterize the behavior of model gradients, with stability quantifying whether the input is a local optimum, and consistency quantifying disagreement between gradients at different locations in the space.

Finally, considering that the interpretation of image attributions depends heavily on their semantic interpretation, we propose a new procedure for visualizing attributions which addresses several concerns associated with the visualization methods commonly employed in the past. Notably, we address the problems of artificial introduction of information from reference inputs, loss of color channel-specific information, and loss of attribution sign.

Using our new visualization procedure, we present our proposed measures qualitatively evaluated on ImageNet [8] using gradients from a pre-trained ResNet-34 model, as illustrated in Fig. 1 with evaluation examples shown in Fig. 2. In summary, our contributions include:

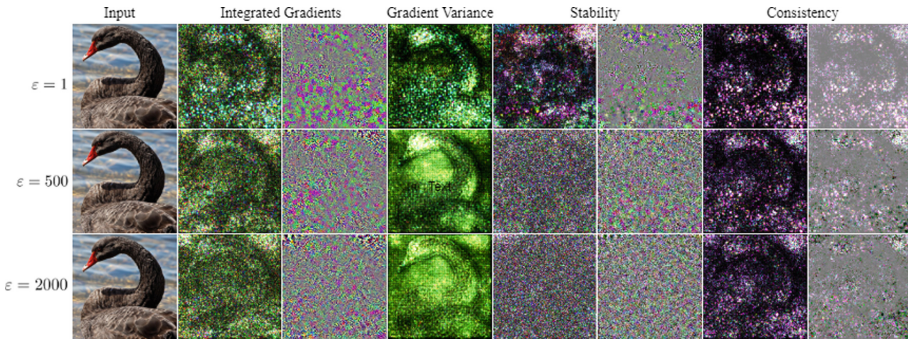
- A method of more accurately and faithfully visualizing attributions
- A mathematical formulation to describe and develop a generalized family of novel integrated attribution measures
- Several specific useful measures of interest constructed from descriptive statistics using our formulation



**Fig. 1.** Illustration of our method as compared with Expected Gradients [10]. We notably include a locality parameter as well as the ability to compute additional descriptive statistics beyond a simple sample mean.

## 2 Related Work

Early methods in explainability, such as layer-wise relevance propagation (LRP) [3], decompose the predictions of nonlinear classifiers to obtain attributions for individual pixels. Many current methods utilize various forms of gradient information in order to generate attributions [1, 2]. In an effort to increase the robustness of these feature attributions, Sundararajan et al. [27] selected a set of axioms to guide the development of a more robust attribution measure which they call integrated gradients. Integrated gradients are computed by taking a linear path from an input of interest to a baseline input, and integrating the gradients of the model with respect to the input over this path, as is discussed in greater detail below in Sect. 3.2. To allow for efficient computation of integrated gradients, Hesse et al. [14] consider a special class of nonnegatively homogenous deep neural networks, and to remove the arbitrary baseline selection issues associated with integrated gradients. With their iterated integrated attributions [5], Barken et al. utilize linear interpolations of the input as well as intermediate representations from within the model. Erion et al. [10] use examples from the training dataset as baselines, which re-contextualizes the resulting attribution values as the expectation of model gradients over the data, with similar approach being taken by Lundberg et al. [19] to approximate Aumann-Shapley (SHAP) values. Merrill et al. define a “generalized integrated gradients” [21] from an axiomatic, algebraic perspective in the context of Aumann-Shapley values in order to extend the concept of path-integrated credit assignment to more diverse function spaces such as those relevant to applications in finance. While we also define a “generalized integrated gradients” in this work, ours is instead framed in the context of developing a broader family of integrated attribution measures



**Fig. 2.** Summary of our newly proposed family of attribution measures and visualization methods [best viewed in color]. Each measure is computed using the locality method of Eq. 4, and the sampling method of expected gradients [10] using 500 sample points (see Fig. 4 for additional sampling details). From top to bottom:  $\varepsilon = 1, \varepsilon = 500, \varepsilon = 2000$ . From left to right: input, local integrated gradients (unsigned), local integrated gradients (signed), gradient variance (unsigned only), stability (unsigned), stability (signed), consistency (unsigned), consistency (signed).

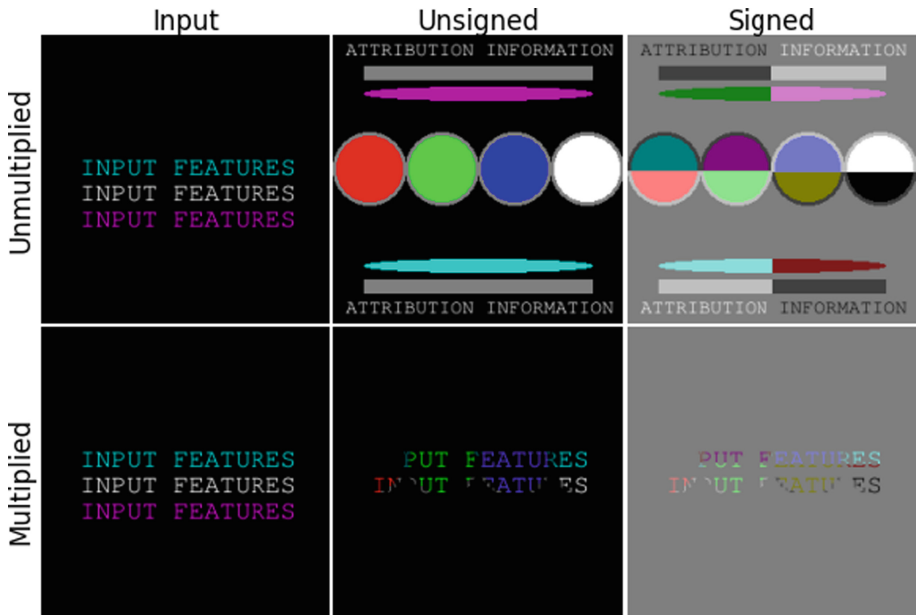
of which the path-integrated gradients is a special case. Extending prior work in attribution to include hidden units within a neural network, Dhamdhere et al. [9] introduce the notion of *conductance*. This neuron attribution builds on the integrated gradients attribution method, with conductance being formulated as the flow of integrated gradients via a given hidden unit. This work on neuron conductance is refined by Shrikumar et al. [26], who develop a scalable implementation they call neuron integrated gradients. In another instance of attribution methods being applied towards other deep learning tasks, Jha et al. [15] construct an attribution-based confidence (ABC) metric for measuring whether an output can be trusted. Variants of the metric utilize different attribution methods, one being integrated gradients. Hase et al. [12] also compare several salience-based explanation methods (such as integrated gradients) and several search-based methods such as their parallel local search. In particular, they posit that the use of out-of-distribution counterfactual inputs like the baselines required for integrated gradients is problematic. Our proposed generalized method builds on the success of expected gradients [10] in addressing the above concerns regarding the out-of-distribution counterfactual inputs which are often used in attribution methods, and enables further development of nuanced attribution measures.

### 3 Generalized Integrated Attributions

#### 3.1 Visualization of Pixel Attributions

We first discuss the approach we have taken for visualizing pixel attributions for computer vision tasks, as this has been an area of significant recent interest [1, 24] and is essential for the accurate interpretation of computed attributions.

Any transformation of attribution values which is not invertible will result in loss of information by compression, as will any transformation which introduces information from an outside source in the form of noise. Previous methods, such as integrated gradients [27] and expected gradients [10], chose to visualize computed attribution values by taking the absolute value (compression), aggregating values for each color channel to a single per-pixel attribution (compression), clipping extreme values (compression), scaling to the range [0, 1], and then multiplying the resulting values by the original input image (noise). Perhaps most importantly, multiplication by the input results in an extremely misleading attribution visualization which artificially resembles the original input image (see Fig. 3). Furthermore, the choice of aggregating color channels needlessly obscures channel-dependent information, which demonstrate to be highly informative. While clipping to quantiles and rescaling to a given range may often be necessary to produce visualizations perceptible to human users, we should always make careful note of these transformations and remind ourselves that each of these transformations may reveal or obfuscate unique sources of information.



**Fig. 3.** Comparison of visualization methods [best viewed in color]. We consider a hypothetical input (column 1) and a hypothetical attribution consisting of a test pattern with both positive and negative values to illustrate the difference between signed and unsigned approaches. We can observe that multiplying by the input results in a significant loss of information and bias towards the input.

**Unsigned Visualization.** When we are interested in the magnitude of attribution values and not whether they are positive or negative, we can take the absolute value of the attributions and scale them to  $[0, 1]$  after first clipping extreme values. This preserves color channel information and introduces no artificial information from the original input. Using this method, attributions with small magnitude are dark while attributions with large magnitude are bright (see row 1, column 2 in Fig. 3).

**Signed Visualization** In contrast to unsigned visualization, if we wish to visualize the difference between positive and negative attribution values, we instead scale the attributions to  $[-1, 1]$  after clipping extreme values. Then, we selectively brighten or darken a blank slate image starting from 50% uniform brightness to obtain the final attribution map. This method preserves both the sign of the attributions and all color-dependent information while introducing no artificial bias from the original input. Using this method, negative attributions are dark while positive attributions are bright (see row 1, column 3 in Fig. 3).

As demonstrated in Fig. 3, there are unique advantages and disadvantages to both signed and unsigned attribution visualization, and ideally both should be used in concert when interpreting attribution results. Importantly, any visualization of attribution measures should not be obscured by any information from a particular reference input unless absolutely necessary, in the interest of introducing as little bias as possible into the final interpretation of a given attribution result. In cases where an unambiguous mask can be constructed from prediction attributions, such a mask might be used to highlight regions of a particular reference input, but this masking should be performed with caution and careful consideration in order to avoid the misinterpretation of input features as attribution results.

### 3.2 Extending Expected Gradients

The reformulation of integrated gradients as an expected value developed by Erion et al. [10] allows the original path integrals of Sundararajan et al. [27] to be completely discarded in favor of volume integrals over the input space. However, this simplification was not thoroughly realized in the presentation of expected gradients. We now reformulate integrated gradients as a generalized integral over a volume in the input space. Sundararajan et al. [27] defines the *path integrated gradients* (Eq. 1) for a model  $F$  and path function  $\gamma(\alpha)$ ,  $\alpha \in [0, 1]$  from the input  $x_0$  to a baseline which we recall below:

$$\text{PathIntegratedGrads}_{\gamma}(x_0) ::= \int_{\alpha=0}^1 \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \quad (1)$$

Erion et al. [10] extends this with the method of *expected gradients*, which aggregates the path integrated gradients for a distribution of many paths  $\gamma$ , and specifically considers a collection of paths using a uniform distribution over examples

from the training set as baseline path endpoints. We now define the *generalized integrated gradients* (Eq. 2) over a set  $\mathbb{S}$  and a probability density function  $p_{\mathbb{S}}$ :

$$\begin{aligned} \text{GeneralizedIntegratedGrads}(\mathbb{S}) &::= \mathbb{E}_{\mathbb{S}} [\nabla F] \\ &= \int_{\mathbb{S}} \nabla F(x) p_{\mathbb{S}}(x) dx \end{aligned} \quad (2)$$

If we follow the method of expected gradients [10] and assume a uniform distribution over  $\mathbb{S}$  with  $|\mathbb{S}|$  the volume (or even more generally the Lebesgue measure) of  $\mathbb{S}$ , we obtain Eq. 3:

$$\text{GeneralizedIntegratedGrads}(\mathbb{S})::= \frac{1}{|\mathbb{S}|} \int_{\mathbb{S}} \nabla F(x) dx \quad (3)$$

The generalized formulation of Eq. 2 includes the expected gradients [10] as a special case, which in turn includes the path-based integrated gradients [27] as a special case. To illustrate an immediate advantage over expected gradients, we define below the *local integrated gradients* (Eq. 4) for a neighborhood  $\mathcal{B}_{\varepsilon}(x_0)$ , i.e. the  $n$ -dimensional ball of radius  $\varepsilon$  centered on an input  $x_0$ , where  $n$  is the number of dimensions of the input, and  $V_n(\varepsilon)$  is the volume of the  $n$ -dimensional ball of radius  $\varepsilon$ . Notice that for  $\varepsilon = \infty$ , this method is equivalent to expected gradients when the space is sampled along paths  $\gamma$  between the input  $x$  and examples from the training dataset, but other volume sampling methods are now available for exploration. Importantly, by controlling the radius  $\varepsilon$ , we are now also able to collect the integrated gradients corresponding to a specific locality (Figs. 4, and 8a), and we can do the same for the other descriptive statistics which we develop below (Figs. 5, 8b, 6, 8c, 7, 8d).

$$\text{LocalIntegratedGrads}(x_0, \varepsilon)::= \frac{1}{V_n(\varepsilon)} \int_{\mathcal{B}_{\varepsilon}(x_0)} \nabla F(x) dx \quad (4)$$

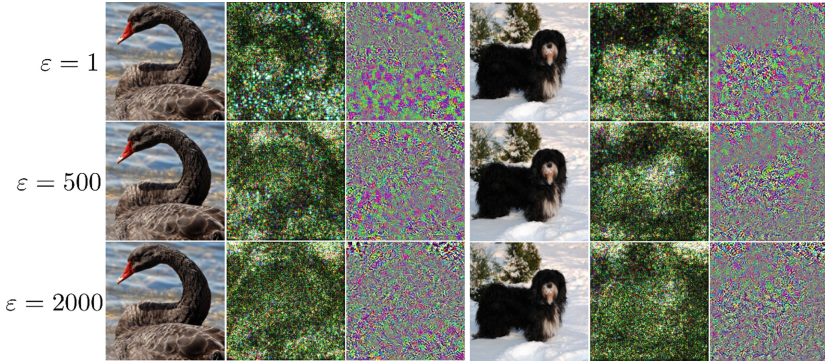
We then to compute a numerical approximation of the desired integral over the desired set. We can again follow the example of expected gradients [10] and collect sample points  $S$  within the set  $\mathbb{S}$  to approximate with a Monte Carlo integral as follows in Eq. 5, where  $|\mathbb{S}|$  is the volume of the set  $\mathbb{S}$ , and  $|S|$  is the number of points in the sample  $S$ .

$$\begin{aligned} \text{GeneralizedIntegratedGrads}(\mathbb{S}) &::= \frac{1}{|\mathbb{S}|} \int_{\mathbb{S}} \nabla F(x) dx \\ &\sim \frac{1}{|\mathbb{S}|} \left[ \frac{|\mathbb{S}|}{|S|} \sum_{s \in S} \nabla F(s) \right] = \frac{1}{|S|} \sum_{s \in S} \nabla F(s) = \mathbb{E}_S [\nabla F] \end{aligned} \quad (5)$$

### 3.3 Novel Attribution Measures

Using the above framework developed for generalizing integrated gradients (Eq. 2), we now propose three new feature attribution measures as descriptive





**Fig. 4.** Local integrated gradients of Eq. 4 [best viewed in color]. We can observe how the choice of  $\varepsilon$  results in noticeably different attributions, and how the unsigned and signed visualizations reveal different patterns especially with respect to color channels. We compute this measure for  $\varepsilon = 1, \varepsilon = 500, \varepsilon = 2000$  (top, middle, bottom row respectively). Immediately to the right of the input are the attributions visualized using our unsigned method. We sample  $\mathcal{B}_\varepsilon(x_0)$  using a reference dataset as in the method of expected gradients [10], using 100 reference elements and 5 uniform random sample points on each of these vectors within the ball  $\mathcal{B}_\varepsilon(x_0)$ , for a total of 500 sample points, yielding a *local expected gradients*.

statistics which account for different aspects of model behavior. Again assuming a uniform distribution over  $\mathbb{S}$ , Monte Carlo approximation with a sample set  $S$  can be applied for each of these measures as easily as for generalized integrated gradients by following the example of Eq. 5. If we follow the method of selecting  $\mathbb{S}$  used for local integrated gradients 4, we can also again compute all of the following measures according to a desired locality radius  $\varepsilon$ .

**Gradient Variance.** Building on the formulation of integrated gradients as a sample mean by Erion et al. [10], we now construct a sample variance (Eq. 6) to quantify the dispersion of model gradients over the set  $\mathbb{S}$ . Note that we again are able to preserve color channel information, but since variances are strictly positive measures, we do not need to consider visualizing negative values (Figs. 5 and 8b).

$$\begin{aligned}
 \text{GradientVariance}(\mathbb{S}) & \\
 & ::= \frac{1}{|\mathbb{S}|} \int_{\mathbb{S}} (\nabla F(x) - \mathbb{E}_{\mathbb{S}} [\nabla F(x)])^2 dx \quad (6) \\
 & = \mathbb{E}_{\mathbb{S}} [\nabla F(x)^2] - \mathbb{E}_{\mathbb{S}} [\nabla F(x)]^2
 \end{aligned}$$

**Stability** We propose a measure of local stability as follows (Eq. 7). For each sample point  $s$  within the set  $\mathbb{S}$ , we compute the vector  $s - x_0$  defining the offset of



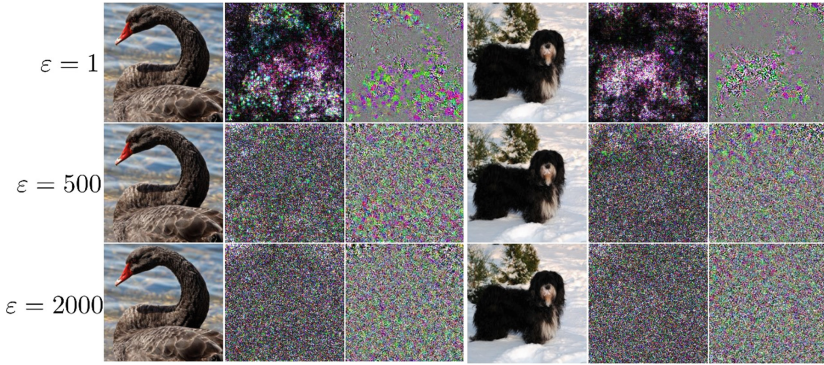
**Fig. 5.** Gradient variance of Eq. 6 [best viewed in color]. We can again observe the effect of the locality radius  $\epsilon$  and the presence of color-dependent patterns. We also obtain visualization which are significantly more salient than those we obtained for local expected gradients (Fig. 4). We use the same sample scheme and choices of  $\epsilon$  as in Fig. 4. Since variances are strictly positive, we only use our unsigned visualization method (Sect. 3.1).

this sample point from the original input. We then compute the cosine similarity of between the offset vector and the gradients at the sample point  $\nabla F(s)$ . The intuition of this measure is that if the gradients at a sample location point back toward the input, then that input can be considered ‘stable’, in that the input is a local optimum. The total stability measure is taken as the expectation of these angles over the set  $\mathbb{S}$  as:

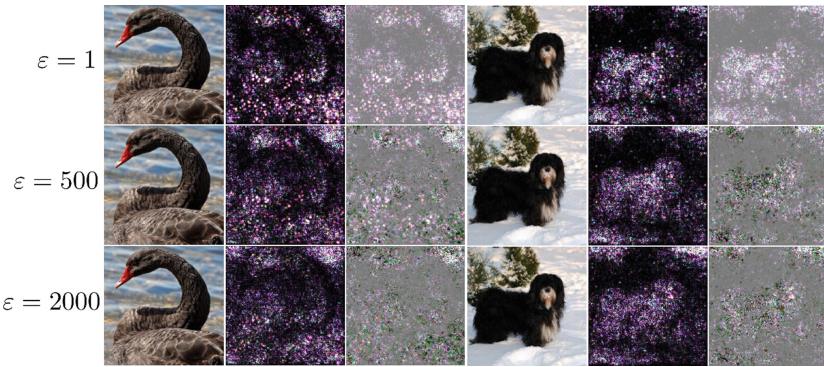
$$\begin{aligned}
 \text{Stability}(\mathbb{S}, x_0) &::= \frac{1}{|\mathbb{S}|} \int_{\mathbb{S}} \frac{(x - x_0) \cdot \nabla F(x)}{\|x - x_0\| \|\nabla F(x)\|} dx \\
 &= \mathbb{E}_{\mathbb{S}} [\cos(\theta)], \\
 &\theta \text{ the angle between } \nabla F(x) \text{ and } (x - x_0)
 \end{aligned}
 \tag{7}$$

To avoid losing channel-dependent information, we compute three angles  $(\theta_{rg}, \theta_{gb}, \theta_{br})$  using pairs of pixels as 2-dimensional vectors. We map the values  $\theta_{rg}$  to the blue channel,  $\theta_{gb}$  to the red channel, and  $\theta_{br}$  to the green channel for Fig. 6.

**Consistency** Finally, we propose a measure which we call ‘consistency’ (Eq. 8). For each sample point  $s$  within the set  $\mathbb{S}$ , we compute the cosine similarity of the gradients of the model at the sample point  $\nabla F(s)$  and the gradients at the input  $\nabla F(x_0)$ . The intuition of this measure is that if the gradients at a sample location point in the same direction as the gradients at the input, then the model gradients are locally consistent with each other. The total consistency measure



**Fig. 6.** Stability measure of Eq. 7 [best viewed in color]. We only observe salient images for small  $\varepsilon$ , as for larger  $\varepsilon$  the input  $x_0$  is likely no longer a local optimum. We use the same sample scheme and choices of  $\varepsilon$  as Fig. 4.

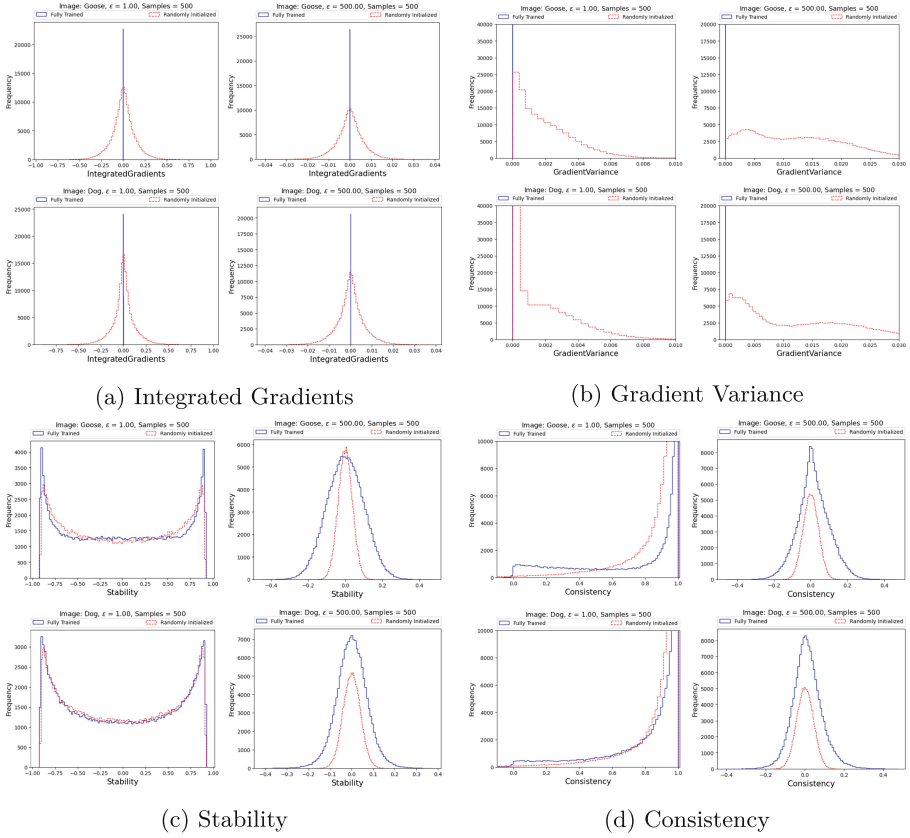


**Fig. 7.** Consistency measure of Eq. 8 [best viewed in color]. This measure allows for determining which pixels the gradients at nearby images  $x \in \mathcal{B}_\varepsilon(x_0)$  either agree or disagree with the gradients at the image  $x_0$ . The same sample scheme and choices of  $\varepsilon$  are the same as in Fig. 4.

is taken as the expectation of these angles over the set  $\mathbb{S}$  as:

$$\begin{aligned}
 \text{Consistency}(\mathbb{S}, x_0) &::= \frac{1}{|\mathbb{S}|} \int_{\mathbb{S}} \frac{\nabla F(x) \cdot \nabla F(x_0)}{\|\nabla F(x)\| \|\nabla F(x_0)\|} dx \\
 &= \mathbb{E}_{\mathbb{S}} [\cos(\theta)], \\
 &\theta \text{ the angle between } \nabla F(x) \text{ and } \nabla F(x_0)
 \end{aligned}
 \tag{8}$$

Again, we preserve the color-dependent information by computing three angles ( $\theta_{rb}, \theta_{rg}, \theta_{bg}$ ) using pairs of pixels as 2-dimensional vectors, and mapping the similarity value representing a given pair of channels to the remaining channel for the final visualization (Fig. 7).



**Fig. 8.** Histograms of each of our novel measures corresponding to gradients from both a randomly initialized and fully-trained ResNet-34 (Row 1: goose, Row 2: dog). We can observe some recognizable parametric families and different paradigms for small and large  $\epsilon$ , with a clear distinction between the trained (blue) and untrained (red) models. If attribution values converge in distribution during model training, this may reveal valuable insight regarding future training optimizations, heuristics, and diagnostics. (Color figure online)

**Generalized Integrated Attributions.** In the interest of describing all of the above measures as well as any similarly constructed descriptive statistic using a single unified formulation, we provide the following definition of a *generalized integrated attribution* (Eq. 9). By selecting an attribution definition  $\mathcal{A}$ , a model  $F$ , a set of interest  $\mathbb{S}$ , and a probability density function  $p_{\mathbb{S}}$ , we can access a limitless number of unique statistics to describe high-dimensional gradient landscapes.

$$\text{GeneralizedIntegratedAttribution}(\mathcal{A}, F, \mathbb{S}, p_{\mathbb{S}}) ::= \int_{\mathbb{S}} \mathcal{A}(F, x) p_{\mathbb{S}}(x) dx \tag{9}$$

Note that we do not necessarily include a particular input  $x_0$  as a required argument, as we can in theory compute attributions over entire sets  $\mathbb{S}$  without referring directly to any single input. For the case of local integrated gradients, the set of interest  $\mathbb{S}$  is the  $\varepsilon$ -ball centered at an input  $x_0$ , but this is a justification for the choice of  $\mathbb{S}$ . Note that our stability and consistency measures appear to require an input  $x_0$ , but these can be framed instead as particular choices of attribution function  $\mathcal{A}$ .

While many interesting attribution measures such as the several new measures we have introduced above are described by the family of generalized integrated attributions, there are likely many more complex attributions of interest which cannot be formulated concisely as a single integral or expected value. Nevertheless, this new formulation can assist in the classification and analysis of newly-developed attribution measures.

## 4 Evaluation Using Quantus [13]

In addition to providing the above qualitative attribution outputs, we also consider a quantitative evaluation of our approach, although there is still no broad consensus regarding reliable metrics for attribution [1, 16, 24]. We provide some

**Table 1.** Quantitative evaluation of novel attribution measure family using the Quantus XAI library [13]. Metrics used are: PixelFlipping [3], FaithfulnessCorrelation [6], MaxSensitivity [28], AvgSensitivity [28], Sparseness [7], Complexity [6]. Results are averaged over the CIFAR10 [18] test set. Our (local) Expected Gradients, Gradient Variance, Stability, and Consistency measures were each computed by Monte Carlo integration using 100 sample points within the ball of radius  $\varepsilon$ .

Method	$\varepsilon$	Faithfulness ( $\uparrow$ )		Robustness ( $\downarrow$ )		Complexity	
		PixFlip	FaithCorr	MaxSens	AvgSens	Sparse( $\uparrow$ )	Complex( $\downarrow$ )
Integrated Gradients [27]	n/a	0.23133	0.04774	0.13018	0.11247	<b>0.59017</b>	<b>6.29801</b>
Saliency [4, 23]		0.28260	0.03239	0.13332	0.11957	0.43868	6.60204
GradientShap [20]		0.23266	0.04752	0.18278	0.14631	0.58966	6.29854
FeatureAblation [17]		0.18525	0.13089	0.11974	0.10510	0.58176	6.32653
FeaturePermutation [11]		0.16536	<b>0.14338</b>	0.19927	0.18554	0.55717	6.38713
Deconvolution [29]		0.30896	-0.00627	<b>1.9e-08</b>	<b>1.8e-08</b>	0.51399	6.48971
Expected Gradients	1	0.24490	0.02238	1.13295	1.03500	0.50759	7.58803
	$10^3$	0.23667	0.01751	1.33943	1.07549	0.46421	7.66907
Gradient Variance	1	<b>0.34443</b>	0.04312	0.78553	0.65590	0.56980	7.41242
	$10^3$	0.27669	0.03585	1.12104	0.80185	0.46126	7.65275
Stability	1	0.28154	0.01003	1.42457	1.25999	0.41586	7.74354
	$10^3$	0.28020	-0.00411	1.02010	0.99323	0.41840	7.74035
Consistency	1	0.27990	-0.00454	0.30972	0.29827	0.10239	7.99951
	$10^3$	0.28233	-0.00418	1.16962	1.09159	0.39830	7.76662



quantitative results in Table 1 using the Quantus XAI library, which provides a toolkit of various attribution methods and evaluation. Metrics in this library are organized into several broad categories such as Faithfulness, Robustness, and Complexity. Given that each metric is unique and sensitive to its own hyperparameters, detailed descriptions defining each method are provided by Hedstrom et al. [13]. We evaluated each attribution method on the full CIFAR-10 [18] test set, using a pre-trained ResNet-18 model.

## 5 Conclusion

In this work, we present a generalized formulation of the feature attribution methods integrated gradients and expected gradients by contextualizing expected values as general integrals over sets of interest. Furthermore, we demonstrate how this approach makes available new sources of attribution information, such as differences between local and nonlocal attribution paradigms, and novel attribution measures. This framework also allows for new forms of parametric control over attribution measures such as the choice of locality radius  $\varepsilon$  and the sampling distribution over the set  $\mathbb{S}$ . Overall, this new formulation of integrated attributions represents a significant transition towards a much broader family of generalizable measures. Additionally, we introduce a novel method for visualizing attributions which addresses information loss in current approaches. Such approaches to more explainable AI can have significant societal impact, enabling better transparency and bias mitigation than treating learning models as black boxes. Our work to reduce misinformation and bias in feature attributions directly addresses the growing need for transparency and fairness with respect to machine learning.

### 5.1 Limitations

Our method depends heavily on Monte Carlo integration, therefore the accuracy, computational efficiency, and robustness of our attribution results likewise depend on the design and incorporation of effective numerical integration schemes. Specifically, for large sets  $\mathbb{S}$ , or equivalently large radius  $\varepsilon$ , the number of sample points required to obtain a good approximation of the true integral increases exponentially. Similarly, any axiomatic properties of our family of measures would also depend on a good approximation of the underlying integral, so this poses a computational challenge to scaling if we desire to measure attributions over large sets. Note however, that other state-of-the-art methods such as expected gradients methods have similar numerical scaling limitations.

**Future Work** Numerical techniques, such as those developed by Mitchell et al. [22], Reeger et al. [25], and Hesse et al. [14], may serve to improve the efficiency and accuracy of integrated attributions. Additionally, we can conduct convergence analyses for hyperparameters such as the sample size and the locality radius  $\varepsilon$ , and we can explore the metrics based on Aumann-Shapley values

developed by Lundberg et al. [19]. In addition, we should assess our new family of measures using an analytic or algebraic approach similar to the selection of desirable axioms by Sundararajan et al. [27] and Merrill et al. [21]. Erion et al. [10] made another significant contribution with their method of using attribution prior for training regularization, so we should apply this technique to train models using our new measures for these attribution priors. To explore additional sources of model attribution, and since integrated gradients forms the basis for layer conductance [26], we should develop implementations of our new measures which can be applied within the space of convolutional filters. Extending attribution measures to applicability in the abstract feature space may also have the benefit of revealing new sources of relevant attribution information.

**Acknowledgements.** This material is based upon work supported by the National Science Foundation under Grant No. 2134237. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: *Advances in Neural Information Processing Systems*, vol. 31 (2018)
2. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: *International Conference on Learning Representations* (2018)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**(7), e0130140 (2015)
4. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. *J. Mach. Learn. Res.* **11**, 1803–1831 (2010)
5. Barkan, O., Elisha?, ., Asher, Y., Eshel, A., Koenigstein, N.: Visual explanations via iterated integrated attributions. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2073–2084 (October 2023)
6. Bhatt, U., Weller, A., Moura, J.M.: Evaluating and aggregating feature-based model explanations. In: *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3016–3022 (2021)
7. Chalasani, P., Chen, J., Chowdhury, A.R., Wu, X., Jha, S.: Concise explanations of neural networks using adversarial training. In: *International Conference on Machine Learning*, pp. 1383–1391. PMLR (2020)
8. 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)
9. Dhamdhere, K., Sundararajan, M., Yan, Q.: How important is a neuron. In: *International Conference on Learning Representations* (2019)
10. Erion, G., Janizek, J., Sturmfels, P., Lundberg, S., Lee, S.I.: Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Mach. Intell.* **3**, 1–12 (2021)

11. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.* **20**(177), 1–81 (2019)
12. Hase, P., Xie, H., Bansal, M.: The out-of-distribution problem in explainability and search methods for feature importance explanations. In: *Advances in Neural Information Processing Systems*, vol. 34 (2021)
13. Hedström, A., et al.: Quantus: an explainable AI toolkit for responsible evaluation of neural network explanations and beyond. *J. Mach. Learn. Res.* **24**(34), 1–11 (2023)
14. Hesse, R., Schaub-Meyer, S., Roth, S.: Fast axiomatic attribution for neural networks. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems*. vol. 34, pp. 19513–19524. Curran Associates, Inc. (2021)
15. Jha, S., et al.: Attribution-based confidence metric for deep neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 32. Curran Associates, Inc. (2019)
16. Jyoti, A., Ganesh, K.B., Gayala, M., Tunuguntla, N.L., Kamath, S., Balasubramanian, V.N.: On the robustness of explanations of deep neural network models: A survey. [arXiv:abs/2211.04780](https://arxiv.org/abs/2211.04780) (2022)
17. Kokhlikyan, N., et al.: Captum: A unified and generic model interpretability library for pytorch. [arXiv preprint:2009.07896](https://arxiv.org/abs/2009.07896) (2020)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)
19. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* **2**(1), 56–67 . <https://doi.org/10.1038/s42256-019-0138-9>
20. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777. NIPS'17, Curran Associates Inc. (2017)
21. Merrill, J., Ward, G., Kamkar, S., Budzik, J., Merrill, D.: Generalized integrated gradients: a practical method for explaining diverse ensembles. [arXiv preprint arXiv:1909.01869](https://arxiv.org/abs/1909.01869) (2019)
22. Mitchell, S.A., Awad, M.A., Ebeida, M.S., Swiler, L.P.: Fast approximate union volume in high dimensions with line samples. Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2018)
23. Morch, N., et al.: Visualization of neural networks using saliency maps. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. vol. 4, pp. 2085–2090 vol.4 (1995). [10.1109/ICNN.1995.488997](https://doi.org/10.1109/ICNN.1995.488997)
24. Rao, S., Böhle, M., Schiele, B.: Towards better understanding attribution methods. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10223–10232 (June 2022)
25. Reeger, J.: Approximate integrals over the volume of the ball. *J. Sci. Comput.* **83** (05 2020). <https://doi.org/10.1007/s10915-020-01231-y>
26. Shrikumar, A., Su, J., Kundaje, A.: Computationally efficient measures of internal neuron importance. *CoRR* [abs/1807.09946](https://arxiv.org/abs/1807.09946) (2018)
27. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *International Conference on Machine Learning*, pp. 3319–3328. PMLR (2017)
28. Yeh, C.K., Hsieh, C.Y., Suggala, A., Inouye, D.I., Ravikumar, P.K.: On the (in) fidelity and sensitivity of explanations. In: *Advances in Neural Information Processing Systems*, vol. 32 (2019)



29. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I*, pp. 818–833. Springer International Publishing, Cham (2014). [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)

# Author Index

## A

Ali, Amin Ahsan 242  
Almalki, Amani 291  
Ayyar, Meghna P. 160

## B

Bailly, Kévin 353  
Banerjee, Ayan 176  
Barddal, Jean Paul 208  
Benois-Pineau, Jenny 160  
Bhaumik, Kishor Kumar 242  
Bonato, Jacopo 1  
Bonnard, Jules 353  
Bora, Revoti Prasad 368

## C

Cai, Jianfei 307, 337  
Calderara, Simone 1  
Chae, Daewon 16  
Chen, Guang 322  
Chen, Shangyu 307, 337  
Chen, Yang 111  
Chen, Zhaojie 32  
Chen, Zixin 322  
Chen, Zixuan 48  
Cotogni, Marco 1  
Cucchiara, Rita 1

## D

Dapogny, Arnaud 353  
De Min, Thomas 64  
de Souza Britto Jr., Alceu 208  
Dhombres, Ferdinand 353  
Dutta, Paramartha 128

## F

Fang, Pengfei 307  
Fernandez, Amanda S. 455

## G

Gao, Han 94  
Guo, Liangxuan 111  
Gupta, Sandeep K. S. 176

## H

Harandi, Mehrtash Tafazzoli 307  
Hu, Chaoshun 48  
Huynh, Viet 337

## J

Jagravi, Istapriya 258  
Jung, Haeji 16

## K

Katsikas, Dimitrios 80  
Kim, Hyunsoo 400  
Kim, Jinkyu 16  
Kim, Jun Hee 400  
Kim, Minha 242  
Kim, Sungyoon 16  
Koerich, Alessandro Lameiras 208

## L

Lai, Jian-Huang 48  
Laiti, Francesco 64  
Latecki, Longin Jan 291  
Laurensi, Israel A. 208  
Lee, Eunjo 400  
Lee, Kangjun 225  
Li, Lening 276  
Liang, Guoqiang 32  
Liberatori, Benedetta 64  
Liu, Lingen 322  
Luo, Huiyuan 94  
Luo, Lei 276

## M

Ma, Lizhuang 192  
Maity, Aranyak 176

Mandal, Sourab 128  
 Mitra, Pabitra 258  
 Moon, Suhong 16  
 Mosconi, Matteo 1  
 Mukherjee, Arnabi 128

**P**

Paidi, Avinash 258  
 Panariello, Aniello 1  
 Pardo, Xosé M. 144  
 Parga, César D. 144  
 Park, Seongbeom 16  
 Park, Seunghyun 16  
 Passalis, Nikolaos 80  
 Patrick, David 455  
 Payne, Ethan 455  
 Peruzzo, Elia 383  
 Phung, Dinh 307, 337  
 Porrello, Angelo 1

**R**

Raja, Kiran 368  
 Ramachandra, Raghavendra 368  
 Regueiro, Carlos V. 144  
 Ricci, Elisa 64

**S**

Sabetta, Luigi 1  
 Sangineto, Enver 383  
 Sebe, Nicu 383  
 Shen, Fei 94  
 Son, Jaeman 400  
 Song, Jihoon 400  
 Song, Yiran 192  
 Sorokin, Andriy 1

Su, Shibin 32  
 Sun, Yanguang 276

**T**

Tefas, Anastasios 80  
 Terhörst, Philipp 368  
 Trivedy, Vivek 291

**V**

Vasilescu, M. Alex O. 420  
 Veldhuis, Raymond 368

**W**

Wang, Yu 439  
 Woo, Simon S. 225, 242

**X**

Xing, Songlong 383

**Y**

Yang, Xiaohao 307  
 Yang, Xingli 439  
 Ye, Biaohua 48  
 Yu, Shan 111  
 Yun, Hyunju 16

**Z**

Zemmari, Akka 160  
 Zhang, Shizhou 32  
 Zhang, Yanning 32  
 Zhang, Zhengtao 94  
 Zhao, He 337  
 Zhao, Xiaoyan 439  
 Zhou, Qianyu 192