

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

LNCS 15326

Pattern Recognition

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

26 Part XXVI

ICPR
2024 INDIA



 Springer

MOREMEDIA 

Lecture Notes in Computer Science

15326

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 XXVI

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 978-3-031-78394-4

ISBN 978-3-031-78395-1 (eBook)

<https://doi.org/10.1007/978-3-031-78395-1>

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

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

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

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

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

If disposing of this product, please recycle the paper.

President's Address

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

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

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

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

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

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

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

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

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

September 2024

Arjan Kuijper
President of the IAPR

Preface

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

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

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

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

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

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

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

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

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

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

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

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

October 2024

Umapada Pal
Josef Kittler
Anil Jain

Organization

General Chairs

Umapada Pal
Josef Kittler
Anil Jain

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

Program Chairs

Apostolos Antonacopoulos
Subhasis Chaudhuri
Rama Chellappa
Cheng-Lin Liu

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

Publication Chairs

Ananda S. Chowdhury
Wataru Ohyama

Jadavpur University, India
Tokyo Denki University, Japan

Competition Chairs

Richard Zanibbi
Lianwen Jin
Laurence Likforman-Sulem

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

Workshop Chairs

P. Shivakumara
Stephanie Schuckers
Jean-Marc Ogier
Prabir Bhattacharya

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

Tutorial Chairs

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

Doctoral Consortium Chairs

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

Organizing Chairs

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

Organizing Committee

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

Finance Chairs

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

Awards Committee Chair

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

Sponsorship Chairs

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

Exhibition and Demonstration Chairs

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

International Liaison, Visa Chair

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

Publicity Chairs

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

Women in ICPR Chairs

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

Event Manager

Alpcord Network

Track Chairs – Artificial Intelligence, Machine Learning for Pattern Analysis

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

Track Chairs – Computer and Robot Vision

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

Track Chairs – Image, Speech, Signal and Video Processing

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

Track Chairs – Biometrics and Human Computer Interaction

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

Track Chairs – Document Analysis and Recognition

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

Track Chairs – Biomedical Imaging and Bioinformatics

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

Metareviewers (Conference Papers and Competition Papers)

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

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

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

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

Reviewers (Competition Papers)

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

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

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

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

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

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

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

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

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

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

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

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

Sneha Shukla
 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	Weiqiang Wang
Umarani Jayaraman	Weiwei Guo
V. S. Malemath	Weixia Zhang
Vallidevi Krishnamurthy	Wei-Xuan Bao
Ved Prakash	Weizhong Jiang
Venkata Krishna Kishore Kolli	Wen Xie
Venkata R. Vavilthota	Wenbin Qian
Venkatesh Thirugnana Sambandham	Wenbin Tian
Verónica Maria Vasconcelos	Wenbin Wang
Véronique Ve Eglin	Wenbo Zheng
Víctor E. Alonso-Pérez	Wenhan Luo
Vinay Palakkode	Wenhao Wang
Vinayak S. Nageli	Wen-Hung Liao
Vincent J. Whannou De Dravo	Wenjie Li
Vincenzo Conti	Wenkui Yang
Vincenzo Gattulli	Wenwen Si
Vineet Padmanabhan	Wenwen Yu
Vishakha Pareek	Wenwen Zhang
Viswanath Gopalakrishnan	Wenwu Yang
Vivek Singh Baghel	Wenxi Li
Vivekraj K.	Wenxi Yue
Vladimir V. Arlazarov	Wenxue Cui
Vu-Hoang Tran	Wenzhuo Liu
W. Sylvia Lilly Jebarani	Widhiyo Sudiyono
Wachirawit Ponghiran	Willem Dijkstra
Wafa Khlif	Wolfgang Fuhl
Wang An-Zhi	Xi Zhang
Wanli Xue	Xia Yuan

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

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

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

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

Contents – Part XXVI

Attention-Guided Energy-Based Model for Out-of-Distribution Data Detection	1
<i>Zongjing Cao, Yan Li, and Byeong-Seok Shin</i>	
Enhancing Fairness and Robustness in Label-Noise Learning Through Advanced Sample Selection and Adversarial Optimization	16
<i>Naihao Wang, YuKun Yang, Haixin Yang, and Ruirui Li</i>	
SEDA: Similarity-Enhanced Data Augmentation for Imbalanced Learning	32
<i>Javad Sheikh, Farshad Farahnakian, Fahimeh Farahnakian, Luca Zelioli, and Jukka Heikkonen</i>	
A Robust Framework for Evaluation of Unsupervised Time-Series Anomaly Detection	48
<i>Onat Gungor, Amanda Rios, Priyanka Mudgal, Nilesh Ahuja, and Tajana Rosing</i>	
The Analog Layer: Simulating Imperfect Computations in Neural Networks to Improve Robustness and Generalization Ability	65
<i>Giovanni Maria Manduca, Antonino Furnari, and Giovanni Maria Farinella</i>	
Improving Out-of-Distribution Data Handling and Corruption Resistance via Modern Hopfield Networks	81
<i>Saleh Sargolzaei and Luis Rueda</i>	
A-FSL: Adaptive Few-Shot Learning via Task-Driven Context Aggregation and Attentive Feature Refinement	97
<i>Riti Paul, Sahil Vora, Nupur Thakur, and Baoxin Li</i>	
Joint Modal Heterogeneous Balance Hashing for Unsupervised Cross-Modal Retrieval	114
<i>Jie Zhang and Mingyong Li</i>	
Interpretable Visual Semantic Alignment via Spectral Attribution	129
<i>Shivanvitha Ambati, Vineet Padmanabhan, Wilson Naik Bhukya, and Rajendra Prasad Lal</i>	

Progressive Learning Based on QP Distance for Enhancing HOP In-Loop Filter	144
<i>Penghao Fu, Cheolkon Jung, Yang Liu, and Ming Li</i>	
Asymmetric Learned Image Compression Using Fast Residual Channel Attention	156
<i>Yusong Hu, Cheolkon Jung, Yang Liu, and Ming Li</i>	
LDINet: Long Distance Imaging Through RGB and NIR Image Fusion	171
<i>Lin Mei, Hao Zhang, and Cheolkon Jung</i>	
Efficient Long-Range Context Modeling for Motion Forecasting with State Space Models	186
<i>Zhiwei Dong, Ran Ding, Jiaxiang Wang, and Wei Li</i>	
DWT-SALF: Subband Adaptive Neural Network Based In-Loop Filter for VVC Using Cyclic DWT	202
<i>Yunfeng Liu and Cheolkon Jung</i>	
Optimal Time Sampling in Physics-Informed Neural Networks	218
<i>Gabriel Turinici</i>	
HeFormer: A Lightweight Transformer Combining Hash Estimation for Link Prediction	234
<i>Teng Sun, Xiaoqiang Xiao, Xu Zhang, and Weixun Ning</i>	
Rethinking Attention Module Design for Point Cloud Analysis	249
<i>Chengzhi Wu, Kaige Wang, Zeyun Zhong, Hao Fu, Junwei Zheng, Jiaming Zhang, Julius Pfommer, and Jürgen Beyerer</i>	
A Hyperparameter Optimization Method Based on Statistical Orthogonal Design for Neural Network Models	268
<i>Yu Wang, Bo Du, Shufan Wu, and Xingli Yang</i>	
NeuroDAVIS-FS: Feature Selection Through Visualization Using NeuroDAVIS	284
<i>Chayan Maitra, Anwasha Sengupta, and Rajat K. De</i>	
S3TC: Spiking Separated Spatial and Temporal Convolutions with Unsupervised STDP-Based Learning for Action Recognition	299
<i>Mireille El-Assal, Pierre Tirilly, and Ioan Marius Bilasco</i>	
Recommendation of Data-Free Class-Incremental Learning Algorithms by Simulating Future Data	315
<i>Eva Feillet, Adrian Popescu, and Céline Hudelot</i>	

Incremental Object 6D Pose Estimation	331
<i>Long Tian, Amelia Sorrenti, Yik Lung Pang, Giovanni Bellitto, Simone Palazzo, Concetto Spampinato, and Changjae Oh</i>	
Enhancing Quantum Diffusion Models with Pairwise Bell State Entanglement	347
<i>Shivalee R. K. Shah and Mayank Vatsa</i>	
Efficient Prescribed-Time and Robust Zeroing Neural Networks for Computing Time-Variant Plural Stein Matrix Equation	362
<i>ShuPeng Li and ZhaoHui Qi</i>	
HRA: Heuristic Reordering Approach for Preserving Dependency in Hierarchical Time Series Forecasting	376
<i>Santosh Palaskar, Surya Shravan Kumar Sajja, Nandyala Hemachandra, and Narayan Rangaraj</i>	
TS-NUC : Nearest Unlike Cluster Guided Generative Counterfactual Estimation for Time Series Classification	392
<i>Ayanabha Ghosh, Rishi Jain, Shubham Parida, and Debasis Das</i>	
Convergence of a L2 Regularized Policy Gradient Algorithm for the Multi Armed Bandit	407
<i>Ştefana-Lucia Aniţa and Gabriel Turinici</i>	
Delving into Feature Space: Improving Adversarial Robustness by Feature Spectral Regularization	423
<i>Zhen Cheng, Fei Zhu, Xu-Yao Zhang, and Cheng-Lin Liu</i>	
Cluster-Mined Negative Samples for Enhanced Unsupervised Sentence Representation Learning	440
<i>Yuhang Zhang, Wenjie Zhang, Yang Hua, Zun Wang, Xiaoning Song, and Xiao-jun Wu</i>	
Author Index	457



Attention-Guided Energy-Based Model for Out-of-Distribution Data Detection

Zongjing Cao¹, Yan Li¹, and Byeong-Seok Shin¹

Department of Computer Science and Engineering, Inha University, Incheon, Korea
zjcao@inha.edu, {leeyeon, bsshin}@inha.ac.kr

Abstract. Detecting out-of-distribution (OOD) data is crucial for the safe and reliable deployment of deep learning models in open-world scenarios. While energy-based models (EBMs) have shown promising potential in OOD detection through the use of an energy function to capture the underlying probability distribution of data, previous approaches have primarily utilized logits or class probabilities from the fully connected layer to compute energy scores. However, logits are inherently class-specific and focus mainly on the relationship between the input and known classes, potentially ignoring the rich information embedded in raw feature representations that are essential for identifying OOD samples. This study introduces a novel approach that utilizes patterns within the feature space to calculate energy scores instead of relying on logits or class probabilities. We propose a spatial attention score to generate class-specific features for each category, which are then used to compute the energy score. Furthermore, we develop a new energy function that transforms these features into energy scores, significantly improving the OOD detection performance of EBMs. In experiments conducted on a Cifar-10 pre-trained ResNet-50, our feature-based energy score method reduced the average false positive rate at a true positive rate of 95% by 5.33% compared to the logits-based approach.

Keywords: Energy-based model · Out-of-distribution detection · Uncertainty quantification.

1 Introduction

Out-of-distribution (OOD) data detection is a critical task in computer vision, enabling deep neural networks (DNNs) to operate safely and reliably in real-world applications. During training, DNNs are typically exposed to a specific data distribution. However, real-world data may contain samples that deviate significantly from this distribution, leading to incorrect model predictions. OOD detection aims to identify these anomalous samples and preventing the model from making wrong or misleading decisions [5, 10, 27]. OOD detection is a challenging task due to the lack of prior knowledge about the distribution of OOD samples, requiring models to discriminate between in-distribution (ID) and OOD data points.

The discriminative approach is a simple and effective strategy for OOD detection [24]. It involves training a discriminative model to classify samples as either ID and OOD samples [5, 25]. Hendrycks et al. introduced the Max-softmax method, a simple approach for OOD detection that utilizes the maximum softmax probability (MSP) as a confidence score [8]. A higher MSP score suggests a higher confidence in the prediction, indicating that the sample is more likely to belong to the ID data. However, the performance of the Max-softmax method suffers from the overconfidence of DMMs, which often assign higher MSP scores to OOD samples [26]. On the other hand, energy-based models (EBMs) have shown great potential in OOD detection due to its ability to estimate the probability distribution over the whole data space through an energy function. By assigning lower energy values to data points that align with the learned distribution and higher energy values to OOD samples, EBMs can effectively discriminate between ID and OOD data. The fundamental principle of EBMs is to design a specific energy function $E_{\theta}(x)$ that assigns a unique scalar value to every input data point x . Previous studies typically use logits from the final fully connected (FC) layer of the DNN to calculate energy scores [6, 13]. However, this approach overlooks the valuable information embedded within the feature space itself, resulting in suboptimal OOD detection performance. Logits, also known as pre-softmax activations, represent the raw scores assigned to each class by a deep learning model before normalization into probabilities. While it provide valuable insights into the decision-making process of the model, it inherently focus on the relationship between the input data and the known classes the model has been trained on. Moreover, logits simply represent a compressed version of the features captured in the feature space.

We argue that the logits represent class-specific probabilities, which may limit its ability to capture subtle OOD patterns. The patters in the feature space contain richer information about data structure and relationships and are more suitable for OOD detection. We propose to directly use the patterns in the feature space to compute the energy scores, rather than using the logits or class probability scores output by the FC layer. To obtain robust features, a spatial attention scoring function was used to generate class-specific features for each category. These class-specific features are then used to calculate the energy score through our designed energy function. We replace the FC layer with multiple 1×1 convolutional layers to capture the spatial structure and information within the feature maps. This is because the simple mapping performed by the FC layers may lose valuable details of the raw features. Furthermore, a new energy function was designed to convert the obtained class-specific features into a single, non-probabilistic scalar energy score. Finally, the OOD detector determines whether an input is ID or OOD by comparing its calculated energy score to a predefined threshold.

The proposed method was evaluated using two ID datasets (Cifar-10 and Cifar-100) and multiple OOD test sets (iSUN, LSUN-crop/resize, Places365, Tiny-ImageNet-crop/resize). Compared with existing methods, the proposed method achieved better OOD detection performance. Specifically, on a Cifar-

10 pre-trained ResNet-50, our method reduced the average false positive rate at a true positive rate of 95% (FPRat95) by 5.33% compared to the logits-based energy score method. Our code is publicly available at github.com/zjcao/ebmOOD.

The main contributions of this study are summarized as follows.

- 1) We proposed employing the patterns in the feature space to compute the energy score instead of using the logits of the FC layer output. The 1×1 convolutional layers are utilized in place of the FC layer to capture the structural information of the patterns.
- 2) A spatial attention scoring function was introduced to generate class-specific features for each category, which was used to calculate the energy score.
- 3) We presented a novel feature-based energy function for EBM to convert the derived class-specific features into energy scores.

2 Related Works

2.1 OOD Detection

OOD data can be referred to as “unknown” or “unseen” data because the model has not encountered this data during the training phase. The primary objective of OOD detection is to identify samples that are not drawn from the training distribution. OOD detection is a challenging task because it requires the model to distinguish between “known” and “unknown” data points without any prior knowledge about the distribution of the “unknown” samples.

A variety of approaches have been proposed by researchers to tackle the problem of OOD detection in recent years. These existing methods can be roughly classified into two types: post-hoc methods and regularization-based methods. Hendrycks et al. proposed to use the MSP score to determine whether an instance is ID or OOD [8]. Although the MSP score method is simple to implement, its performance suffers from the overconfidence problem of DNNs, which often assign high confidence scores for OOD samples. To address the problem of the DNNs generating too high confidence scores in OOD samples, Liang et al. used a scaling parameter to smooth the MSP scores. The scaling parameter was calculated on a separate validation dataset [12]. OOD detection can be viewed as a binary classification task. Vaze et al. found a strong correlation between the detection performance of a classifier on OOD data and its accuracy on the ID set [18]. Simply improving the recognition accuracy of the ID classifier can significantly enhance its OOD detection performance. Cen et al. also suggested that the incorrect predictions of ID samples made by the base classifier severely limit its OOD detection performance [9]. In summary, recent works attempt to differentiate between ID and OOD samples in the feature space by analyzing the differences in feature representations, predicted probabilities or logits, and parameter gradients [15, 20].

2.2 EBM for OOD detection

EBMs are a class of generative models that define a probability distribution over data by specifying an energy function. Unlike generative adversarial networks, EBMs do not require a discriminator. Instead, EBM learn a mapping from data to a scalar energy, where lower energy values correspond to higher probability [4, 13]. Due to its flexibility in modeling complex distributions, EBM has been applied in OOD detection tasks.

Consider a function $E_\theta(x)$, where θ represents the parameters of the function and x is the input data. The output of $E_\theta(x)$ is a scalar value. By applying basic probability theory, we can normalize the scores for all potential inputs as follows:

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z(\theta)} \quad (1)$$

The exponential function guarantees that a non-zero probability is assigned to any possible input [6]. The $E_\theta(x)$ is called the energy function. To ensure that data points with high likelihood correspond to low energy values and vice versa, we employ a negative sign in front of E . The Z_θ is a normalization constant to ensure that the density integrates to one, which can be expressed as:

$$Z(\theta) = \int \exp(-E_\theta(x)) dx \quad (2)$$

The core component of an EBM is the energy function, denoted as $E_\theta(x)$. This function assigns a single, non-probabilistic scalar value to each input point x . Liu et al. proposed employing the output $f(x)$ of a DNN as the free energy function. This can be formulated as:

$$E(x, f) = -\log \sum_i^k \exp(f_i(x)) \quad (3)$$

where $f(x)$ represents a DNN classifier mapping an input x to k real-valued logits [13].

In OOD detection, the energy score serves as a discriminator for the OOD detector $g()$, distinguishing between ID and OOD samples. Higher energy scores indicate OOD inputs, while lower scores correspond to ID inputs. This can be formulated as:

$$g(x, \tau, f) = \begin{cases} 0, & \text{if } -E(x, f) < \tau \\ 1, & \text{if } -E(x, f) \geq \tau \end{cases} \quad (4)$$

where τ is the energy threshold, determined such that a high fraction of ID data (e.g., 95%) is correctly classified

3 Pattern-Based Energy Score for OOD Detection

The EBMs aims to capture the underlying probability distribution of the data. It assigns each data points an energy value that reflects its compatibility with

the learned distribution. Lower energy values indicate higher compatibility, while higher values suggest larger deviations from the distribution. The core of EBMs is to find an energy function that maps each point x in the input space to a scalar. The patterns in the feature space contain richer spatial information and are more suitable for OOD detection. We propose to directly use the patterns in the feature space to compute the energy scores instead of using the logits.

3.1 Connection Between EBMs and Discriminant Models

Inspired by [4] and [13], we observed that the EBMs have potential connections with discriminative model. The discriminative model, also known as conditional probability model, focuses on directly learning the relationship between input data and output labels. The core goal of the discriminative model is to optimize a mapping function to minimize the classification error or maximize the probability of correct prediction. Consider a discriminative model $f(x)$ that maps an input x to k real-valued numbers, known as logits. The logits can be converted into a categorical probability distribution using the Softmax function:

$$p(y | x) = \frac{p(y, x)}{p(x)} = \frac{\exp(f_i(x))}{\sum_j^k \exp(f_j(x))} \quad (5)$$

where $f_i(x)$ denotes the i -th element of $f(x)$, corresponding to the logit associated with the i -th class label. By connecting Eq. 1 and Eq. 5, Liu et al., proposed using the denominator of the Eq. 5 to represent the free energy function $E(x, f)$ over $x \in \mathbb{R}^D$, expressed in Eq. 3 [13].

In object detection tasks, datasets often suffer from class imbalance. To capture dataset-specific statistics during uncertainty regularization, Du et al. adds a learnable parameter w to energy function, which can be expressed as:

$$E(x, f; \theta) = -\log \sum_i^k w_k \exp(f_i(x)) \quad (6)$$

where $f_k((x, b); \theta)$ is the logit output for class k in the classification branch [4]. However, this parameter w must be learned from both the ID and OOD data during training phase.

3.2 Feature-Based Energy Score

The logits are derived through the final FC layer of a classification network, which are specifically designed to distinguish between the trained classes. These layers may not be sensitive to patterns within the feature space that deviate from the known classes. OOD samples, by the problem setting of OOD detection task, represent data that falls outside the training distribution. These unseen patterns critical for identifying OOD data might be overlooked by a purely logits-based approach. We propose to directly use the patterns in the feature space to compute the energy score instead of using the logits of the FC layer output. However, there

are two challenges to be solved: **a)** preserving the classification capabilities of the model, and **b)** converting patterns to energy scores.

To address challenge a) we propose to replace the FC layer of the backbone network with the 1×1 convolutional layer. The FC layer processes the entire flattened feature map, thus losing the spatial information preserved in the convolutional layers. The spatial structure of features is important for distinguish ID and OOD data. On the other hand, the 1×1 convolutional layer preserves spatial information by applying a kernel to each position in the feature map.

Given an input image x , a backbone network $\phi()$ extracts feature maps $\mathbf{x} \in \mathbb{R}^{h \times w \times d}$, where h , w , d represent the height, width, dimension of the feature maps respectively. This process can be mathematically expressed as:

$$\mathbf{x} = \phi(x, \theta) \quad (7)$$

where θ denotes the parameters of the backbone network $\phi()$. For example, for an image of size 32×32 , after feature extraction, the shape of its feature map is $4 \times 4 \times 2048$. Subsequently, several different 1×1 convolutional layers are used to generate score tensors.

To address challenge b). Simply, the score tensors of the last 1×1 convolutional layer output can be used to calculate the energy score. However, the design of energy function is very important for EBM, which directly affects the energy score of the input data and indirectly affects the performance of OOD detection. We propose to use the patterns in the feature space to calculate the energy score instead of using the logits of FC layer output. The quality of the energy score is thus influenced by both the feature map and the energy function. Inspire by human cognitive process attention mechanisms have become a powerful tool in many computer vision tasks. To enable the model to selectively focus on feature maps, a spatial attention scoring was proposed to generate class-specific features for each category.

Assume \mathbf{x} is the feature maps of a input data x in feature space, with shape is $h \times w \times d$, which can be decoupled as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{h \times w}$ ($\mathbf{x} \in \mathbb{R}^d$), the classifier (1×1 convolutional layer) for i -th class is assumed to $m_i \in \mathbb{R}^d$. We can define the spatial attention scoring α for the i -th class and j -th location as:

$$\alpha_j^i = \frac{\exp(\mathbf{x}_j m_i)}{\sum_k^{h \times w} \exp \mathbf{x}_k m_i} \quad (8)$$

The α_j^i can be viewed as the probability that class i appears at position j . Finally, the class-specific features f^i for the i -th class:

$$f^i = \sum_k^{h \times w} \left(\frac{1}{h \times w} + \lambda \alpha_k^i \right) \mathbf{x}_k \quad (9)$$

where λ is a hyperparameter used to control the scale of spatial features. Based on above equation our energy score can be formulated as:

$$E(x, \phi, \alpha, \lambda) = -\log \sum_i^k \exp(\alpha(\phi_i(x)) + \lambda \alpha(\phi_i(x))) \quad (10)$$

where the $\phi()$ is a backbone network used to get the feature maps \mathbf{x} of input x .

4 Experimental Results

The proposed method was implemented and evaluated using PyTorch, a popular open-source deep learning framework. Training and validation experiments were performed on an Ubuntu Linux 22.04.1 workstation equipped with four high-performance NVIDIA GeForce RTX 3090 graphics processing units.

4.1 Experimental Setup

ID and OOD datasets. The Cifar-10 and Cifar-100 datasets were used as the ID datasets to train the backbone classifier. The proposed method was evaluated using multiple OOD datasets, including: iSUN [21], LSUN [23], Places365 [14], SVHN [16], Textures [2], and TinyImageNet [3]. These OOD datasets encompass various image sources and domains, offering a thorough assessment of ability of the model to identify OOD instances. Table 1 summarizes the key characteristics of these datasets, including the number of classes, number of images, and data types.

Table 1. Summary of the ID and OOD datasets.

Datasets	Description	# Images	# Classes
Cifar-10	80 million images subset	60,000 ($10 \times 6,000$)	10
Cifar-100	80 million images subset	60,000 (100×600)	100
iSUN	SUN subset	8,925	24
LSUN	Large-scale scenes	10,000 (subset)	10
Places365	Scene recognition	10,000 (subset)	365
SVHN	Street-view house numbers	26,032	10
Textures	Describable textures	5,640 (47×120)	47
TinyImageNet	ImageNet subset	100,000 (200×500)	200

Implementation details. A modified ResNet-50 architecture served as the backbone network for extracting fine-grained features from the input images. To preserve more detailed features, the kernel size of the initial convolutional layer was reduced from 7×7 to 3×3 , and the Maxpool layer was removed. The backbone network was initialized using ImageNet-1K pre-trained weights and then fine-tuned on the Cifar-10 and Cifar-100 training sets using cross-entropy loss. The parameters of the backbone network were updated using the SGD optimizer, configured with a momentum of 0.9 and a weight decay of 0.005.

Metrics for OOD detection. OOD detection performance was evaluated using the following metrics: 1) FPRat95 curve, 2) area under the receiver

operating characteristic (AUROC) curve, and 3) area under the precision-recall (AUPR) curve. These metrics are commonly used to assess OOD detection capabilities [19, 22].

4.2 Experimental Results

Test set classification accuracy. The backbone network was initially fine-tuned on the Cifar-10 and Cifar-100 training sets, following the standard training protocol detailed in previous section. Then, the classification performance of the backbone network was evaluated on the Cifar-10 and Cifar-100 test sets. Table 2 summarizes the experimental results achieved by our proposed method on these two datasets. On Cifar-10 and Cifar-100 test sets, the proposed method achieved classification accuracy (ACC) of 95.40% and 80.07% respectively.

Table 2. Performance comparison of ACC, ECE, MCE and RMSCE of backbone networks on Cifar-10 and Cifar-100 test sets. Higher (\uparrow) and lower (\downarrow) values are better. All values are percentages.

Datasets	ID test set accuracy(%)				
	ECE (\downarrow)	MCE (\downarrow)	RMSCE (\downarrow)	ACC (\uparrow)	Mean Confi.
Cifar-10	1.97	3.01	3.42	95.40	97.28
Cifar-100	8.70	10.28	20.64	80.07	88.76

For reference, we also reported the average confidence, expected calibration error (ECE), maximum calibration error (MCE), and root mean square calibration error (RMSCE) to assess the confidence calibration of the backbone network [22]. Confidence calibration in DNNs refers to the process of aligning the predicted probability of a DNN model with its actual accuracy. Our backbone network achieved an ECE of 1.97% and 8.70% on Cifar-10 and Cifar-100, respectively. The average confidence scores were slightly overconfident at 97.28% and 88.76% on Cifar-10 and Cifar-100, respectively.

OOD detection performance results. The experimental results for OOD detection using the proposed method are presented. The backbone network was first trained on the Cifar-10 and Cifar-100 training sets. Then, both Cifar-10 and Cifar-100 test sets were used as the ID data. Finally, the OOD detection performance was evaluated on various OOD datasets, including: iSUN, LSUN, Places365, SVHN, Textures, and TinyImageNet using AUROC, AUPR and FPRat95 as evaluation metrics. Table 3 summarizes the experimental results achieved by the proposed method on two ID sets and six OOD test sets. Our method achieved average FPRat95 scores of 30.52% and 49.99%, and average AUROC scores of 92.35% and 85.43% on several OOD test sets, respectively.

Comparison with existing methods. To assess how well our suggested method for detecting OOD data works, we compared it with various advanced OOD detection methods, including: ODIN [11], Max-Softmax [1], RMD [17],

Table 3. OOD detection performance on six different OOD datasets using Cifar-10 and Cifar-100 as ID datasets. Higher (\uparrow) and lower (\downarrow) values are better. All values are percentages.

ID datasets	OOD datasets	OOD detection performance			
		AUROC(\uparrow)	AUPR-in(\uparrow)	AUPR-out(\uparrow)	FPRat95(\downarrow)
Cifar-10	iSUN	93.41	90.60	94.39	26.46
	LSUN-crop	97.42	97.62	97.25	13.23
	LSUN-resize	93.41	91.87	93.71	26.64
	Places365	89.33	95.96	73.03	43.47
	SVHN	82.28	90.22	67.25	57.79
	Textures	95.72	93.21	97.34	20.90
	TinyImageNet-c	95.88	95.96	95.58	19.91
	TinyImageNet-r	91.35	89.99	91.32	35.80
	Average	92.35	93.17	88.73	30.52
Cifar-100	iSUN	85.40	82.48	87.08	52.14
	LSUN-crop	93.08	93.03	93.24	31.41
	LSUN-resize	87.99	86.68	88.63	46.91
	Places365	79.30	91.58	56.79	65.52
	SVHN	75.08	85.58	60.31	66.03
	Textures	86.28	77.26	91.60	51.25
	TinyImageNet-c	92.07	91.59	92.42	33.96
	TinyImageNet-r	84.29	82.64	84.95	54.84
	Average	85.43	86.35	81.87	49.99

Entropy-Based, Open-Max [1], Max-Logit [7], Energy-Logit [13]. The performance was evaluated using two metrics: AUROC and FPRat95. Table 4 summarizes the experimental results achieved by the proposed method on several OOD test sets. The detailed report of each method on multiple OOD test sets can be found in Table 5. The experimental results show that our method achieves a reasonable balance between AUROC and FPRat95, indicating that it can effectively detect OOD samples while minimizing false positives. On a Cifar-10 pre-trained ResNet-50, our method reduces the average FPRat95 by 5.33% compared to the logits-based energy score method, 14.16% lower than Max-Softmax method.

4.3 Ablation Study

Visualization of t-SNE. Visualizing features in the feature space by performing a t-SNE projections can help users gain insight into the decision-making process of DNNs [15, 22]. To analyze the decision boundaries of the model, we employed t-SNE to visualize the feature representations of the Cifar-10 and Cifar-100 test sets. Fig. 1 shows the comparison of t-SNE between logits-based method ((a)(c)) and our proposed method ((b)(d)) on the Cifar-10 and Cifar-100

Table 4. Comparison of OOD detection results with existing methods on six OOD datasets using Cifar-10 as the ID datasets. Higher (\uparrow) and lower (\downarrow) values are better. All values are percentages.

Methods	OOD detection performance			
	AUROC(\uparrow)	AUPR-in(\uparrow)	AUPR-out(\uparrow)	FPRat95(\downarrow)
ODIN	82.07	81.33	80.06	52.75
Max-softmax	84.92	83.84	84.12	44.68
RMD	85.19	83.43	83.18	46.26
Entropy-based	87.33	87.22	85.64	43.67
Open-max	89.64	87.76	87.91	36.88
Max-logit	90.46	90.02	88.72	35.56
DICE	90.57	90.20	88.59	35.58
Energy-logit	90.68	90.35	88.83	35.85
Ours	92.35	93.17	88.93	30.52

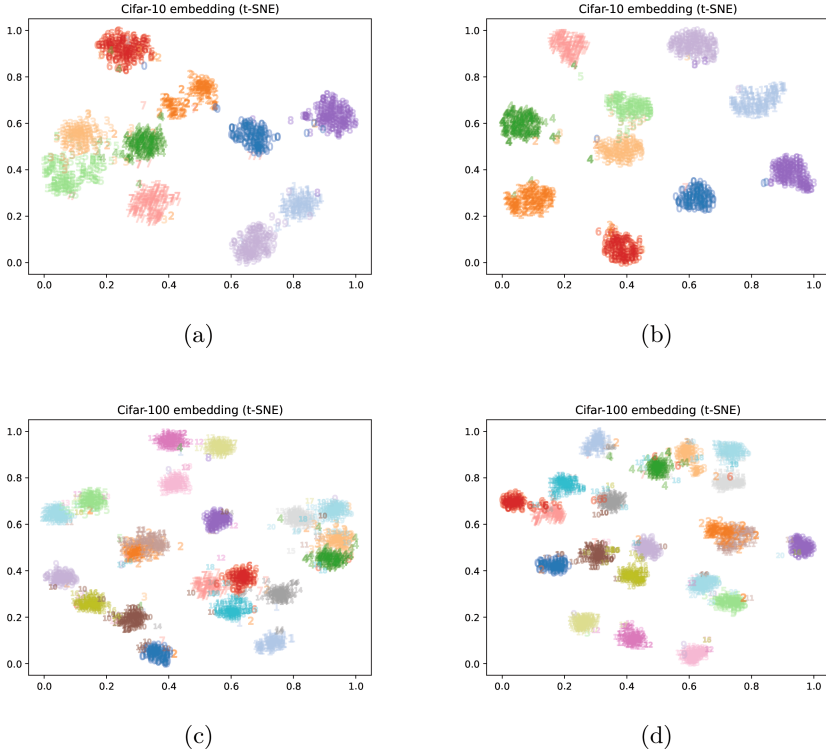


Fig. 1. Comparison of t-SNE embedding between logits-based method (left) and our proposed method (right) on Cifar-10 and Cifar-100 test sets.

Table 5. Comparison of OOD detection performance with existing methods on different OOD test sets. All values are percentages.

Methods	OOD datasets	OOD detection performance			
		AUROC(\uparrow)	AUPR-in(\uparrow)	AUPR-out(\uparrow)	FPRat95(\downarrow)
ODIN	iSUN	82.71	77.24	86.14	51.42
	LSUN	86.22	83.59	88.10	43.37
	Place365	78.48	91.18	58.77	61.00
	SVHN	74.53	85.61	60.02	66.23
	Textures	82.38	70.11	89.66	53.91
	TinyImageNet	83.02	79.66	84.86	51.33
Max-softmax	iSUN	83.57	79.31	87.02	47.73
	LSUN	87.47	85.44	89.49	38.63
	Place365	82.50	93.22	66.00	51.79
	SVHN	84.65	91.91	76.09	45.98
	Textures	85.13	72.98	91.89	43.38
	TinyImageNet	84.27	81.21	86.48	45.63
RMD	iSUN	86.19	81.98	88.17	45.14
	LSUN	88.25	84.71	89.94	38.38
	Place365	82.99	93.20	63.19	55.01
	SVHN	79.46	86.51	69.31	54.24
	Textures	85.77	71.89	92.08	44.96
	TinyImageNet	85.31	82.22	86.39	46.98
Entropy	iSUN	86.61	84.27	88.69	47.31
	LSUN	93.82	92.97	94.42	25.58
	Place365	87.87	95.68	71.01	46.96
	SVHN	89.63	94.59	83.22	34.98
	Textures	89.06	80.00	93.81	40.12
	TinyImageNet	89.83	88.40	90.51	38.94
Open-max	iSUN	90.89	86.91	92.27	35.68
	LSUN	93.22	91.12	94.01	26.42
	Place365	87.49	95.01	70.97	46.74
	SVHN	85.60	89.55	78.47	41.25
	Textures	88.04	75.08	93.32	40.03
	TinyImageNet	89.33	86.66	90.12	39.26
Max-logit	iSUN	91.21	89.16	92.53	35.05
	LSUN	93.47	92.46	94.17	26.45
	Place365	87.66	95.48	71.03	46.89
	SVHN	89.37	94.24	83.09	34.63
	Textures	89.23	80.35	93.92	38.38
	TinyImageNet	89.64	87.99	90.42	38.31
DICE	iSUN	91.07	88.98	92.25	35.72
	LSUN	93.65	92.64	94.27	25.46
	Place365	87.35	95.38	69.45	48.13
	SVHN	89.64	94.60	83.39	34.17
	Textures	89.82	81.28	94.28	37.40
	TinyImageNet	89.67	88.03	90.40	39.17
Energy-logit	iSUN	91.58	89.77	92.73	35.67
	LSUN	93.82	92.97	94.42	25.58
	Place365	87.87	95.68	71.01	46.96
	SVHN	89.63	94.59	83.22	34.98
	Textures	89.06	80.00	93.81	40.12
	TinyImageNet	89.83	88.40	90.51	38.94
Ours	iSUN	93.41	90.60	94.39	26.46
	LSUN	95.41	94.75	95.48	19.93
	Place365	89.33	95.96	73.03	43.47
	SVHN	82.28	92.22	67.25	57.79
	Textures	95.72	93.21	97.34	20.90
	TinyImageNet	93.62	92.98	93.45	27.86

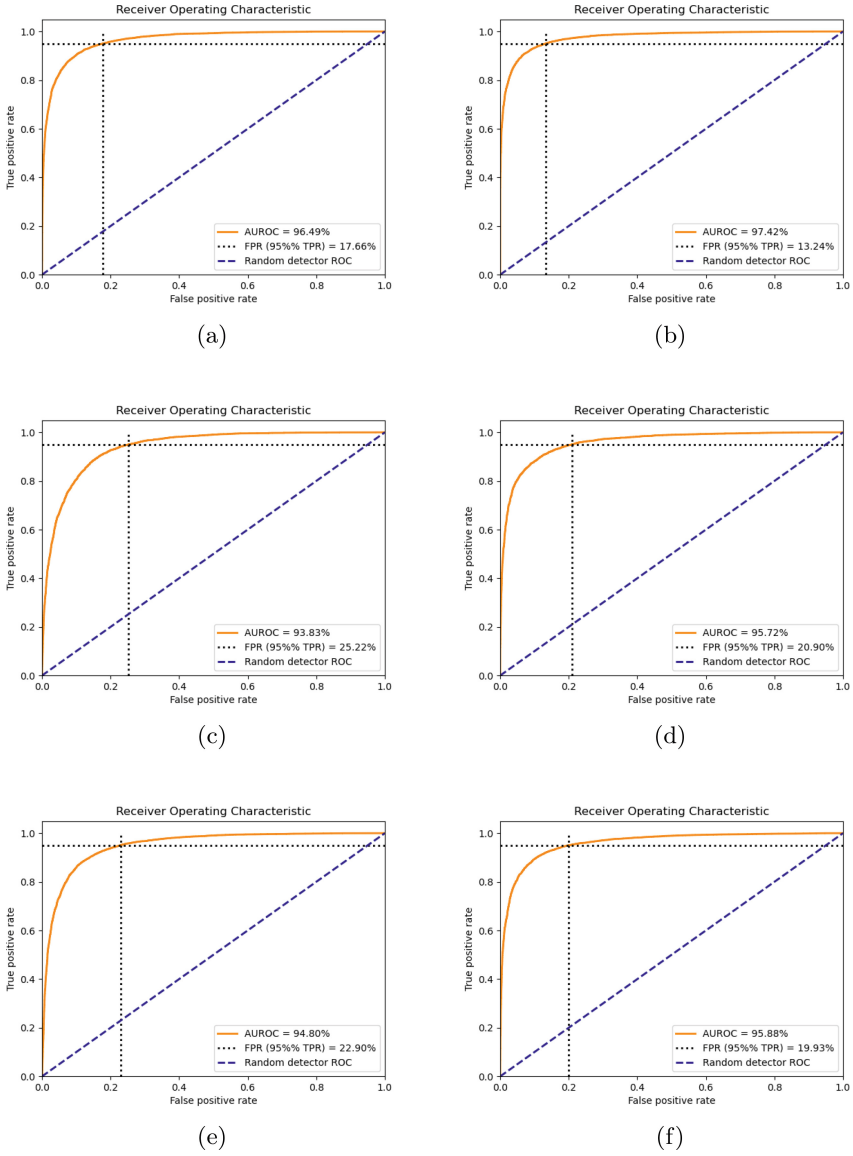


Fig. 2. Comparison of AUROC and FPR curves between logits-based method (left) and our proposed method (right) on the LSUN-crop ((a)(b)), Textures ((c)(d)), TinyImageNet-crop ((e)(f)) as OOD test sets.

test sets. Note that since Cifar-100 has 100 categories, it is not convenient to display all the categories, instead, we randomly display 21 categories. In these figures, numbers of different colors represent different categories. We observed

that each class in our proposed method ((b)(d)) forms a unique cluster in the t-SNE space compared to logits-based method (a)(c).

Visualization of AUROC and FPR curves. The AUROC and FPR curves are commonly used to visualize the performance of OOD detection models. AUROC is a popular metric for evaluating the performance of OOD detection models. The AUROC curve intuitively visualizes the performance of the OOD detection model at different thresholds. The FPR curve, also known as the false alarm curve, plots the FPR against the TPR for various classification thresholds. Fig. 2 presents a comparison of AUROC and FPR curves between logits-based method ((a)(c)(e)) and our proposed method ((b)(d)(f)) on the LSUN-crop, Textures, TinyImageNet-crop OOD test sets. Note that an AUROC value of 1.0 represents perfect classification, while an AUROC of 0.5 suggests random guessing. Generally, a higher AUROC value indicates superior model performance.

5 Conclusion

OOD detection plays a critical role in ensuring the safe and reliable deployment of deep learning models. DNNs are typically trained based on the closed-world assumption, that is, during the training phase the model is only exposed to data from a specific data distribution. In real-world scenarios, the test samples are often drawn from a data source that has a semantic shift from the training set. This poses a significant challenge as it requires the model to distinguish between ID and OOD samples without any prior distribution knowledge of OOD samples.

In this work, we propose a novel approach that directly use the patterns in the feature space to compute the energy scores instead of using the logits. We replace the FC layer with multiple 1×1 convolutional layers to capture the spatial structure and information of the feature maps. Because the simple mapping performed by the FC layers may lose valuable details of the raw features. A spatial attention scoring function was used to generate a class-specific feature for each category. The obtained class-specific features are then used to calculate the energy score through the energy function we designed. Furthermore, a new energy function was designed to convert the obtained class-specific features into energy score, which is a single, non-probabilistic scalar. We evaluated the proposed method on several OOD benchmark datasets and demonstrate its superior OOD detection compared to existing state-of-the-art methods.

Future works include expanding the evaluation of our method to cover a broader range of deep learning tasks, and exploring other energy functions to further enhance detection performance.

Acknowledgements. This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. NRF-2022R1A4A1033549) and Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS2022-00155915, Artificial Intelligence Convergence Innovation Human Resources Development [Inha University]).

References

1. Bendale, A., Boulton, T.E.: Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1563–1572 (2016)
2. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014)
3. 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)
4. Du, X., Wang, Z., Cai, M., Li, Y.: Vos: Learning what you don’t know by virtual outlier synthesis. In: International Conference on Learning Representations (2022)
5. Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A., Triebel, R., Jung, P., Roscher, R., et al.: A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **56**(Suppl 1), 1513–1589 (2023)
6. Grathwohl, W., Wang, K.C., Jacobsen, J.H., Duvenaud, D., Norouzi, M., Swersky, K.: Your classifier is secretly an energy based model and you should treat it like one. In: International Conference on Learning Representations (2019)
7. Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., Song, D.: Scaling out-of-distribution detection for real-world settings. In: International Conference on Machine Learning. pp. 8759–8773. PMLR (2022)
8. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: International Conference on Learning Representations (2016)
9. Jun, C., Luan, D., Zhang, S., Pei, Y., Zhang, Y., Zhao, D., Shen, S., Chen, Q.: The devil is in the wrongly-classified samples: Towards unified open-set recognition. In: The Eleventh International Conference on Learning Representations (2022)
10. Kirchheim, K., Filax, M., Ortmeier, F.: Pytorch-ood: A library for out-of-distribution detection based on pytorch. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4351–4360 (2022)
11. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems* **31** (2018)
12. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. In: International Conference on Learning Representations (2018)
13. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. In: *Advances in neural information processing systems*. vol. 33, pp. 21464–21475 (2020)
14. López-Cifuentes, A., Escudero-Vinolo, M., Bescós, J., García-Martín, Á.: Semantic-aware scene recognition. *Pattern Recogn.* **102**, 107256 (2020)
15. Lu, H., Gong, D., Wang, S., Xue, J., Yao, L., Moore, K.: Learning with mixture of prototypes for out-of-distribution detection. In: International Conference on Learning Representations (2024)
16. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 7. Granada, Spain (2011)
17. Ren, J., Fort, S., Liu, J., Roy, A.G., Padhy, S., Lakshminarayanan, B.: A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022* (2021)

18. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Open-set recognition: A good closed-set classifier is all you need. In: International Conference on Learning Representations (2021)
19. Wang, Q., Fang, Z., Zhang, Y., Liu, F., Li, Y., Han, B.: Learning to augment distributions for out-of-distribution detection. *Advances in Neural Information Processing Systems* **36** (2024)
20. Wang, Y., Mu, J., Zhu, P., Hu, Q.: Exploring diverse representations for open set recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5731–5739 (2024)
21. Xu, P., Ehinger, K.A., Zhang, Y., Finkelstein, A., Kulkarni, S.R., Xiao, J.: Turkergaze: Crowdsourcing saliency with webcam based eye tracking. arXiv preprint [arXiv:1504.06755](https://arxiv.org/abs/1504.06755) (2015)
22. Yang, J., Zhou, K., Li, Y., Liu, Z.: Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision* pp. 1–28 (2024)
23. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint [arXiv:1506.03365](https://arxiv.org/abs/1506.03365) (2015)
24. Yuan, Y., He, R., Dong, Y., Han, Z., Yin, Y.: Discriminability-driven channel selection for out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26171–26180 (2024)
25. Zhang, J., Fu, Q., Chen, X., Du, L., Li, Z., Wang, G., Han, S., Zhang, D., et al.: Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In: The Eleventh International Conference on Learning Representations (2022)
26. Zheng, H., Wang, Q., Fang, Z., Xia, X., Liu, F., Liu, T., Han, B.: Out-of-distribution detection learning with unreliable out-of-distribution sources. *Adv. Neural. Inf. Process. Syst.* **36**, 72110–72123 (2023)
27. Zheng, H., Wang, Q., Fang, Z., Xia, X., Liu, F., Liu, T., Han, B.: Out-of-distribution detection learning with unreliable out-of-distribution sources. *Advances in Neural Information Processing Systems* **36** (2024)



Enhancing Fairness and Robustness in Label-Noise Learning Through Advanced Sample Selection and Adversarial Optimization

Naihao Wang¹, YuKun Yang¹, Haixin Yang², and Ruirui Li¹(✉)

¹ Beijing University of Chemical Technology, Beijing, China
liruirui@mail.buct.edu.cn

² Peking University, Beijing, China

Abstract. The prevalence of label noise in datasets poses significant challenges in supervised learning frameworks, where models that overfit noisy labels experience degraded generalization performance. Numerous robust learning methodologies have been proposed to mitigate the adverse effects of noisy labels on models. Among these methodologies, sample selection-based approaches have garnered notable attention for their promising results on real-world noisy datasets. However, current research predominantly aims at improving the overall accuracy of models in the presence of label noise, often overlooking the critical aspect of fairness across different classes. In this paper, we argue that ensuring model fairness is as crucial as maintaining robustness in the context of label noise. We propose a novel approach that combines advanced sample selection and adversarial optimization to enhance both fairness and robustness simultaneously. Our methodology introduces implicit regularization to model label noise and proposes a sample selection strategy based on the distribution of noise probabilities and associated loss values. Furthermore, we decouple representation learning from classification head learning by leveraging adversarial optimization, focusing on the gradients of the worst-case classification hyperplane. Experimental comparisons on both synthetic and real-world noisy datasets demonstrate that our proposed method achieves superior performance and optimal class fairness. The effectiveness of our approach is substantiated by empirical results, and we provide comprehensive evaluations detailing the robustness against label noise. The code implementing our methodology will be made publicly available at <https://github.com/wangnaihao/DNLL.git>, facilitating further research and development in the field.

This work was supported in part by National Natural Science Foundation of China under Grant No. 62101021.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_2.

1 Introduction

Supervised training relies on well-annotated, high-quality samples. Despite the advent of large foundation models, high-quality annotations remain crucial for fine-tuning to achieve robust and generalizable models. However, label noise persists, and data collected through methods like web scraping or crowdsourcing often includes some erroneous labels. Even manual labeling procedures can inadvertently introduce mislabeling due to varying levels of expertise. Recent research has highlighted the significant impact of data annotation quality on both the generalization capabilities and training efficiency of models. The field of Learning with Noisy Labels (LNL) [7, 11, 13] aims to mitigate the negative effects of incorrect annotations and has garnered significant attention [23].

Before fitting noisy samples, over-parameterized deep neural networks undergo an initial learning phase, focusing exclusively on clean samples. Sample selection methods typically leverage the knowledge gained during this phase to choose high-quality samples and refine the model through pseudo-labels generated from model predictions. Superior models can identify better samples, which further enhances their performance. By employing this feedback loop, sample selection-based methods incrementally improve the testing accuracy of models in the presence of label noise. However, model predictions can sometimes be incorrect, generating erroneous pseudo-labels for unlabeled samples and introducing new false labels. Training the network on these incorrect labels accumulates errors, resulting in confirmation bias. RobustLR [2] corroborates the presence of confirmation bias and identifies a notable proportion of incorrect pseudo-labels in sample selection-based methods. To mitigate this issue, RobustLR introduces a novel confidence estimation technique during the pseudo-label generation process. Addressing instance-dependent noise, CC [29] introduces a centrality and consistency sample selection method to handle significant intra-class variations, while DISC [13] incorporates an instance-aware dynamic threshold for sample selection.

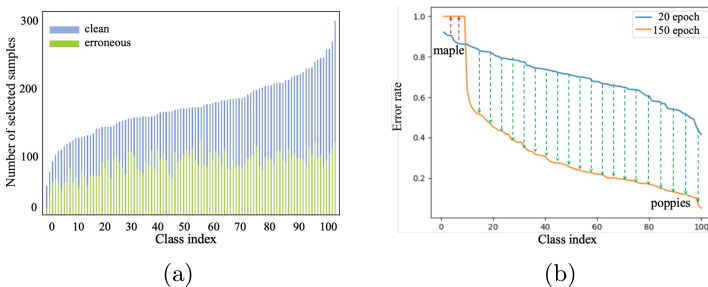


Fig. 1. (a) The distribution of selected samples for different categories after training for 20 epochs. (b) The distribution of error rates for different categories after training for 20 and 150 epochs.

However, previous studies have not adequately addressed the fairness between categories during the training process. Due to the uneven distribution of samples in terms of quantity and learning difficulty, the model’s learning capabilities and prediction errors vary significantly across different categories. Unless additional measures are taken to counteract the disparities introduced by this imbalance, the uneven sample selection and model training can ultimately lead to model bias. In other words, samples that are initially poorly represented at the tail-end of the distribution are likely to become increasingly poorly represented, or even entirely misclassified. To better observe and analyze this phenomenon, we employ the latest DISC method on CIFAR-100 with 60% IDN noise for a series of experiments. Figure 1 illustrates the visual results of the intermediate data from the experiments. Figure 1a depicts the number of samples selected and the proportion of correctly labeled samples at the 20th epoch. It is observed that the categories with a higher number of selected samples also have a higher proportion of correctly labeled samples. Figure 1b shows the model’s test errors across different categories at the 20th and 150th epochs for the same set of experiments. It is evident that the disparity in test errors is dramatically amplified during the training process. Specifically, a few disadvantaged categories, such as the maple class, are almost entirely misclassified, with errors reaching 100%. Therefore, the imbalance between categories is magnified during training, leading to the generation of more erroneous pseudo-labels, exacerbating the model’s confirmation bias, and deteriorating its generalization capability. We believe that ensuring fairness in the model training process is equally important as enhancing the model’s robustness to label noise.

To address the issues mentioned above, in this paper, we strive to build a more effective sample selection strategy and aim to separate representation learning from the learning of the classification head. By adopting a novel perspective on representation optimization, we seek to enhance the fairness of the model. To this end, we propose a new definition of the worst-case classification boundary under an enhanced sample selection strategy and a novel adversarial representation optimization method based on this definition. This method effectively optimizes the representations, thus achieving a larger classification margin. Additionally, to improve sample selection, this paper uses an implicit regularization term to fit the noise probability of labels and performs sample selection by jointly considering the loss distribution and noise distribution. We validate the effectiveness of the proposed method on synthetic noise datasets as well as on real-world noisy datasets. Our approach surpasses the performance of state-of-the-art methods, such as CC[29] and DISC[13], and significantly enhances fairness in learning across different categories. The main contributions of this paper are summarized as follows:

- We propose a novel sample selection method based on the joint consideration of network prediction probabilities and noise-fitting probabilities.
- We introduce a new definition and calculation method for the worst-case classification boundary under label noise.

- We present an adversarial representation optimization method based on the worst-case classification boundary in the presence of label noise.

2 Related Work

2.1 Learning from Noisy Labels

Early studies predominantly focused on developing robust loss functions, such as Peer Loss [18], to enhance noisy label learning. The contemporary approach largely involves sample selection methods. A significant contribution in this field is DivideMix [11], which categorizes samples with minimal losses as clean and considers them labeled data, while treating the remaining samples as unlabeled data. This method employs the MixMatch [1] semi-supervised technique to create pseudo-labels, thereby training a robust classifier. Various studies address confirmation bias by devising improved sample selection strategies [4, 5, 7, 25] or label correction methods [2, 6, 14]. For example, DISC [13] adopts a dual-view approach with an instance-specific dynamic threshold to segregate the dataset into clean, hard, and noisy subsets, handling each accordingly. In this paper, we introduce a dynamic, instance-specific sample selection strategy based on implicit regularization-based noise estimation, which demonstrates higher precision in identifying accurate samples compared to existing methods.

2.2 Implicit Regularization

Implicit regularization can be regarded as a statistical method for sparsity, playing a role similar to minimizing L_1 loss in sparse noise learning and is currently utilized in various models [30]. Among these methods, one noteworthy approach is SOP [15], which is closely related to our method. SOP leverages implicit regularization for noisy label learning, providing a sparse representation of the residual term between the prediction and the observed noisy label. An enhancement of SOP, known as SOP+, incorporates semi-supervised learning techniques. Nevertheless, SOP+ does not comprehensively account for the potential biases that may emerge throughout the evolving training procedure.

3 METHOD

We propose the Debaised Noisy Label Learning (DNLL) framework, based on the concept of the worst-case decision plane. The overall framework is illustrated in Figure 2, and the pseudo-code is provided in Appendix A. The framework comprises three core modules: Noise Estimation with Implicit Regularization, Enhanced Selection, and Reverse Optimization. Additionally, the framework integrates a supervised loss with random Mixup [17] augmentation and consistency regularization using both strong and weak augmentations. DNLL leverages the FixMatch [20] semi-supervised framework to generate and utilize pseudo-labels. An implicit regularizer estimates noise, preventing the model from

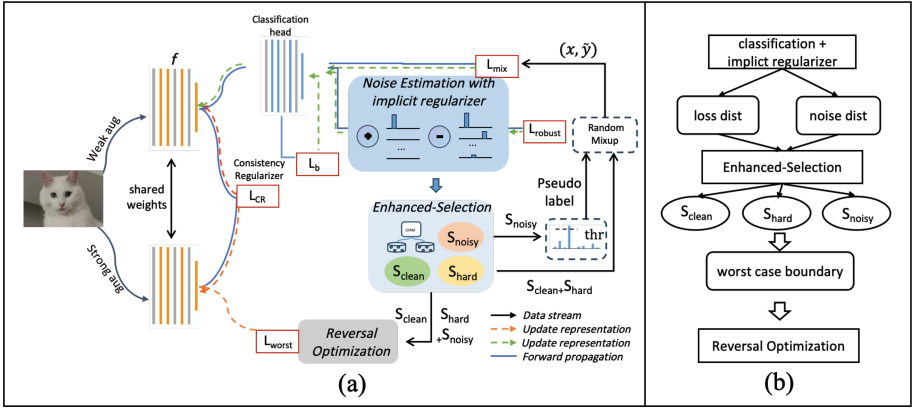


Fig. 2. The framework of the proposed DNLL includes (a) the network architecture and (b) the data workflow.

fitting incorrect labels. We also introduce a novel sample selection strategy that combines noise probability and loss distributions from a dual-perspective approach. Consequently, DNLL dynamically classifies samples into three sets during training: the clean set, the hard set, and the noisy set. The clean set stresses high precision in sample selection, establishing reliable margins among categories and addressing the imbalanced representations caused by the worst-case scenarios defined by the hard and noisy sets. To achieve this, DNLL employs a gradient reversal layer to compute adversarial loss and subsequently updates network parameters for features in the latent space. Detailed discussions of these modules follow in the subsequent subsections.

For convenience, we initially frame the problem as a classification task, denoted by C as the number of categories. The data set is defined as $D = \{(x_i, y_i)\}_{i=1}^N$, where x_i represents the i -th sample, and y_i denotes its corresponding noisy label. Here, θ and ψ are the parameters for the network and the latent space, respectively. The dataset size is represented by N . The sets S_{clean} , S_{hard} , and S_{noise} represent the clean, hard, and noisy data subsets, respectively. The parameters c , n , and h indicate the sizes of these sets. The function $f()$ indicates the network’s prediction, and $L()$ represents the Cross-Entropy (CE) loss.

3.1 Noise Estimation with Implicit Regularization

Previous research has found that over-parameterized structures, represented by deep neural networks, have implicit preferences for low-rank and sparse solutions. Recent study [15] assumes that noise is sparse and thus it can be modeled in a low-rank pattern. Inspired by this idea, this paper introduces an implicit regularization term to recover label noise during the learning process. Specifically, we represent the real correct labels in the following form:

$$\tilde{y}_i = y_i - \beta_i^2 \cdot y_i + \gamma_i^2 \cdot (1 - y_i). \quad (1)$$

In this context, y_i represents the one-hot encoding of the given label of sample i . $\beta_i \in [0, 1]$ represents the sparse format of noise probability. $\gamma_i \in [0, 1]$ represents the sparse format of the corrected label probability. y_i and $(1 - y_i)$ are orthogonal vectors, so as are β_i and γ_i . Under this transformation, learning under noisy labels becomes minimizing the difference between $f(x_i; \theta)$ and \tilde{y}_i and L_{robust} can be simplified by Equation.1 to:

$$\min_{\theta, \{\beta_i, \gamma_i\}_{i=1}^N} \frac{1}{N} \sum_{i=1}^N L(f(x_i; \theta) + \beta_i^2 y_i - \gamma_i^2 (1 - y_i), y_i). \quad (2)$$

Parameters β_i and γ_i , which are learnable for each sample, play a crucial role in modeling sparsity to estimate noise probabilities and correct label probabilities. As demonstrated in [16], initializing the vectors β_i and γ_i with very small positive values allows for the recovery of probability values in sparse vectors through implicit regularization during the over-parameterized model learning process. Throughout the training, the learning of parameter γ_i tends to be more cautious, lagging behind the learning of the network parameters θ and noise parameters β_i . This is because ineffective updates to θ and β_i would render the updates to γ_i futile. To prevent premature learning of γ_i before $f(x_i; \theta)$ and β_i have been properly trained, our research employs a two-stage learning strategy, consisting of a warm-up phase followed by an ongoing learning phase. During the warm-up phase, γ_i is approximated to be nearly zero, thereby simplifying Equation 2 to:

$$\min_{\theta, \{\beta_i, \gamma_i\}_{i=1}^N} \frac{1}{N} \sum_{i=1}^N L(f(x_i; \theta) + \beta_i^2 y_i). \quad (3)$$

During the warm-up phase, the network predominantly concentrates on the preliminary learning of the function f and the estimation of the noise probability β_i . Specifically, we apply the Cross-Entropy loss to update the network parameters θ , and the Mean Squared Error (MSE) loss to refine the parameters β_i . The following formula conducts the updates:

$$\theta \leftarrow \theta - \tau_\theta \cdot \frac{\partial L_{ce}(\theta, \beta_i)}{\partial \theta}. \quad (4)$$

$$\beta_i \leftarrow \beta_i - \tau_\beta \cdot \frac{\partial L_{mse}(\theta, \beta_i)}{\partial \beta_i}, i = 1, \dots, N. \quad (5)$$

During the continual learning phase, the parameter γ_i initiates its learning process using the pre-established parameters θ and β_i . Throughout this stage, a multitasking approach is employed to integrate all other relevant losses into the learning process. The Mean Squared Error (MSE) loss function is utilized to facilitate the update of γ_i . The update rule for γ_i is given by the following formula:

$$\gamma_i \leftarrow \gamma_i - \tau_\gamma \cdot \frac{\partial L_{mse}(\theta, \beta_i)}{\partial \gamma_i}, i = 1, \dots, N. \quad (6)$$

Please note that γ_i cannot be updated using the Cross-Entropy loss (CE loss), as the partial derivative of L_{ce} with respect to γ_i is always equal to 0.

This innovative two-stage design allows DNLL to effectively estimate the probability of label noise during the initial warm-up phase. By doing so, it significantly mitigates the risk of the model fitting to noisy labels, thereby enhancing the overall reliability and accuracy of predictions in the subsequent learning phase.

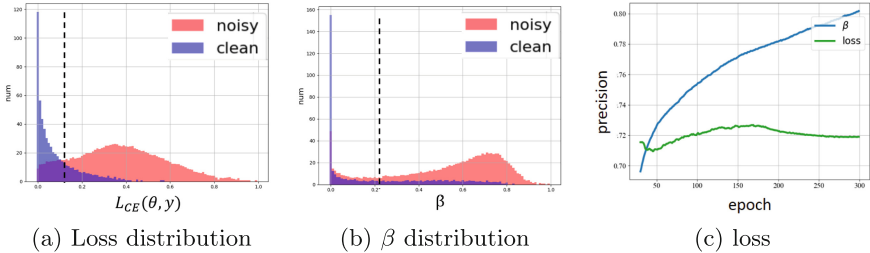


Fig. 3. Comparison between loss and β . (a) The distribution of noise and clean data in loss. (b) The distribution of noise and clean data in β .

3.2 Enhanced-Selection

By leveraging implicit regularization, we can determine the probabilities of label noise and confidence scores for the samples. Based on their distributions, we have formulated a dynamic sample selection strategy called Enhanced-Selection. This approach necessitates both an exceptionally clean sample set to identify category margins and a diverse sample set for data augmentation and training. Consequently, rather than simply dividing the dataset into clean and noisy subsets, Enhanced-Selection dynamically partitions the dataset into three subsets: clean, hard, and noisy.

Based on the proposed sample selection strategy, we first use the Gaussian Mixture Model (GMM) to fit the distribution of β_i and employ a threshold of $\sigma = 0.5$ to select a clean sample set. Next, we apply the GMM to fit the distribution of the losses, using the same threshold to select hard samples, excluding those from the clean set. The remaining samples with high predicted losses are identified as noisy samples. Figures 3a and 3b present the histogram distributions of the losses and noise probabilities, respectively, under a 60% IDN noise ratio in the CIFAR-100 dataset. Due to overlapping loss distributions between clean and noisy samples, a strategy that only selects samples with lower losses results in reduced accuracy. The distribution of noise probabilities helps separate clean samples from noisy ones, thereby improving selection accuracy.

We also plotted the precision of the selected clean set throughout the entire training process, as illustrated in Figure 3c. Compared to the precision of the clean set chosen using the small-loss strategy, our method demonstrates significantly higher precision that consistently increases, ultimately exceeding 80%.

3.3 Latent Compression by Reversal Optimization

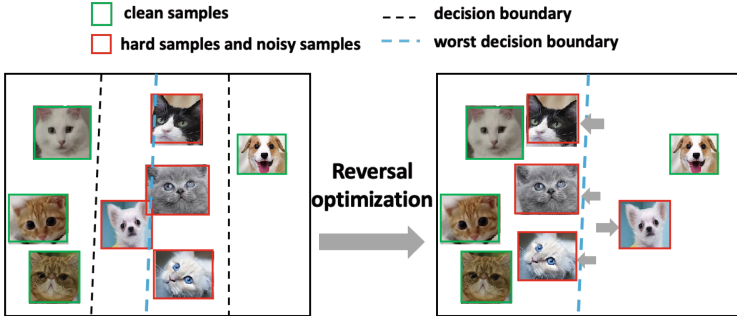


Fig. 4. Left: Modelling the worst-classifying plane based on the partition of the clean set and the other. Right: Optimizing the representation by reverse-engineering the worst-classifying plane.

Samples that are easier to learn typically incur smaller losses. Consequently, sample selection strategies that prioritize low-loss samples may result in an imbalanced sample distribution. This imbalance tends to worsen during subsequent pseudo-label generation, culminating in model biases and unfairness across various classes. To address this issue, we propose decoupling the learning of representations from the learning of the classification head. We have developed a reversal optimization module, which consists of a reverse gradient optimization layer and an auxiliary classification head. The auxiliary classification head operates in the inner loop to simulate worst-case scenarios, while the reverse gradient optimization layer functions in the outer loop, computing loss gradients relative to the worst-case classification plane. We define this "worst-case classification plane" as the one that optimally discriminates S_{clean} while maximally misclassifying S_{hard} and S_{noisy} .

Let h' represent the classification header and ψ denote the parameters only for the latent space. L_{other} represents the classification loss on the of S_{hard} and S_{noise} , while L_{clean} represents the classification loss on the set of S_{clean} . The worst-case classification plane can be obtained by optimizing the following formula:

$$h_{worst}(\psi) = \arg \max_{h'} (L_{other}(\psi, h') - L_{clean}(\psi, h')). \quad (7)$$

The worst-classification plane $h_{worst}(\psi)$ aims to minimize the distance between all samples and labels in S_{clean} , and simultaneously maximize the distance between samples and labels in sets S_{hard} and S_{noisy} . This operation increases the margin between two class boundaries, as Figure 4 shows.

Subsequently, we further optimize the loss under the worst-classification plane through gradient descent to minimize it:

$$L_{worst} = \min (L_{other}(\psi, h') - L_{clean}(\psi, h')), \quad (8)$$

where:

$$L_{\text{clean}}(\psi, h') = \frac{1}{c} \sum_{i=1}^c L_{ce}(f(\psi, h', x_i), y_i)$$

$$L_{\text{other}}(\psi, h') = \frac{1}{h+n} \sum_{i=1}^{h+n} L_{ce}(1 - f(\psi, h', x_i), y_i).$$
(9)

By optimizing representations in reverse to align with decision boundaries under worst-case scenarios, a superior representation can be achieved. This process reduces biases introduced by data imbalances and the use of pseudo-labels.

3.4 Overall Loss Function

In addition to the implicit regularizer and the loss in the reverse optimization module, the DNLL framework incorporates two additional losses. Firstly, there is a supervised learning loss with RandomMix data augmentation[17], defined by L_{mix} . Secondly, there is a consistency regularization term L_{cr} [27].

Our overall loss function can be represented as follows, and λ_w, λ_{cr} are hyper-parameters:

$$L_{total} = L_{robust} + L_{mix} + \lambda_w L_{worst} + \lambda_{cr} L_{cr}. \quad (10)$$

4 EXPERIMENTS AND ANALYSIS

In this section, we validate the effectiveness of our method on both synthetic noise (using CIFAR-10 and CIFAR-100 [10] datasets with instance-dependent noise) and real-world noise (using Animals-10N [21], Clothing-1M [26], and Web-Vision [12] datasets). Subsequently, we conduct ablation experiments to verify the efficacy of each component. Additional results on symmetric noisy data are included in Appendix C to provide a comprehensive understanding of DNLL. All experiments were conducted using a single GeForce RTX 3090 GPU and implemented with PyTorch 1.8.0.

We evaluate the model’s generalizability using overall accuracy (ACC) and its balance across different categories using average accuracy (mACC). For experiments involving instance-dependent and real-world noise, we utilized results from the literature and followed the representation format specified in those sources for mean and variance.

4.1 Synthetic Noise

Dataset CIFAR-10 and CIFAR-100 are widely used image classification datasets in computer vision. Based on the dependency between data and class labels, existing synthetic label noise can be categorized into two types: Class-Dependent Noise and Instance-Dependent Noise (IDN). IDN is sampled from a truncated Gaussian distribution by setting a random noise rate for each instance. Following prior works[2, 13], we conduct experiments on IDN (20%, 40%, 60%) on both CIFAR-10 and CIFAR-100 datasets.

Table 1. Comparison with the SOTA methods on CIFAR-10 and CIFAR-100 with IDN.

DataSet	CIFAR-10			CIFAR-100		
Method	Inst.20%	Inst.40%	Inst.60%	Inst.20%	Inst.40%	Inst.60%
CE	83.93 ± 0.15	67.64 ± 0.26	43.83 ± 0.33	57.35 ± 0.08	43.17 ± 0.15	24.42 ± 0.16
Co-teaching	88.87 ± 0.24	73.00 ± 1.24	62.51 ± 1.98	43.30 ± 0.39	23.21 ± 0.57	12.58 ± 0.58
Co-teaching+	89.80 ± 0.28	73.78 ± 1.39	59.22 ± 6.34	41.71 ± 0.78	24.45 ± 0.71	12.58 ± 0.58
JoCoR	88.78 ± 0.15	71.64 ± 3.09	63.46 ± 1.58	43.66 ± 1.32	23.95 ± 0.44	13.16 ± 0.91
Reweight-R	90.04 ± 0.46	84.11 ± 2.47	72.18 ± 2.47	58.00 ± 0.36	43.83 ± 8.42	36.07 ± 9.73
Peer Loss	89.12 ± 0.76	83.26 ± 0.42	74.53 ± 1.22	61.16 ± 0.64	47.23 ± 1.23	31.71 ± 2.06
DivideMix	93.33 ± 0.14	95.07 ± 0.11	85.50 ± 0.71	79.04 ± 0.21	76.08 ± 0.35	46.72 ± 1.32
CORSES2	91.14 ± 0.46	83.67 ± 1.29	77.68 ± 2.24	66.47 ± 0.45	58.99 ± 1.49	38.55 ± 3.25
CAL	92.01 ± 0.12	84.96 ± 1.25	79.82 ± 2.56	69.11 ± 0.46	63.17 ± 1.40	43.58 ± 3.30
CC	93.68 ± 0.12	94.97 ± 0.09	94.95 ± 0.11	79.61 ± 0.19	76.58 ± 0.25	59.40 ± 0.46
DISC	96.48 ± 0.04	95.94 ± 0.04	95.05 ± 0.05	80.12 ± 0.13	78.44 ± 0.19	69.57 ± 0.14
DNLL	96.92 ± 0.12	96.59 ± 0.08	96.05 ± 0.08	82.64 ± 0.21	81.33 ± 0.03	75.54 ± 0.04

Experimental Setup We use PreResNet18[9] as the backbone network to train 300 epochs on CIFAR-10 and CIFAR-100. For a fair comparison, we use SGD as the optimizer and set the batch size to 128. We utilized two hyperparameters, namely λ_w, λ_{cr} , which were consistently set as 0.3, and 0.9, respectively, across all experiments. Different learning rate settings were employed when learning different parameters on different datasets, and we have summarized them in Appendix B.

Comparison With SOTA Methods Table 1 compares the performance of various methods under IDN noise. We achieve the best performance on CIFAR-100, outperforming the second-ranked method DISC by 2.52%, 2.89%, and 5.97% at noise ratios of 20%, 40%, and 60% respectively. On CIFAR-10, we are also the best in terms of performance at different noise ratios, outperforming the second-ranked method DISC by 0.44%, 0.65%, and 1% at noise ratios of 20%, 40%, and 60% respectively.

4.2 Real-World Noise

Dataset The Animals-10N dataset consists of 5 pairs of easily confusable images, with a total of 50,000 training images and 5,000 testing images. The noise rate in this dataset is approximately 8%. WebVision is a dataset composed of images from Google and Flickr, containing 1,000 different categories and a total of about 2.4 million images. Following the approach mentioned in the reference, we use the first 50 classes of WebVision as the training data and then test the model using the validation sets provided by both WebVision and ILSVRC2012. Clothing1M is a dataset collected from online shopping websites, consisting of 1 million training samples and 10,000 testing samples. The noise rate in this dataset is approximately 38.5%.

Experimental Setup For Animals-10N, following[3], we used VGG19[19] (not pre-trained) as the backbone network and set the batch size to 64. For Web-Vision, following[22], we used InceptionResNetV2[24] (not pre-trained) as the backbone network and set the batch size to 32. For Clothing1M, following[28], we used ResNet50 (pre-trained)[8] as the backbone network and set the batch size to 64. We conducted unified training for 300 epochs with the same hyperparameter settings as the experiments on CIFAR-10 and CIFAR-100. The learning rate configuration can also be found in Appendix B.

Comparison With SOTA Methods The comparison results between the proposed method and the state-of-the-art (SOTA) methods on real datasets are shown in Table 2. DNLL achieves the best test accuracy on all four datasets. It ties with the CC method for first place on the Clothing1M dataset, outperforming the second-ranking PGDF by 0.21 percentage points. The Clothing1M dataset has a long-tail distribution of samples across categories, which can lead to data imbalance and self-training imbalance during the pseudo-label generation process. The outstanding performance of DNLL on this dataset indicates that it effectively handles this imbalance. On WebVision, DNLL outperforms the second-ranking RobustLR by 0.07 and 0.12 percentage points in TOP1 and TOP5 accuracy, respectively. On ILSVRC12, DNLL surpasses the second-ranking DISC by 0.46 percentage points in TOP1 accuracy. While DISC also proposes dynamic instance selection for sample training, it does not pay as much attention to the data and self-training biases introduced by the selection. In terms of the same dataset’s TOP5 metric, the second-ranking method is CC, which has a test accuracy of 93.76%, 0.55 percentage points lower than our proposed method. We also achieve the best results on the Animals-10N dataset, surpassing DISC, RobustLR, and SSR by 2.1%, 0.7%, and 0.7%, respectively.

Table 2. Comparison with the SOTA methods on Clothing1M and WebVision.

Dataset	Clothing1M	WebVision		ILSVRC12		ANimals 10N
		top1	top5	top1	top5	
CE	69.21	-	-	-	-	-
ELR+	74.81	77.78	91.68	70.29	89.76	-
DivideMix	74.76	77.32	91.64	75.20	90.84	-
PGDF	75.19	81.47	94.03	75.45	93.11	-
CC	75.4	79.36	93.64	76.08	93.86	-
SSR	74.83	80.92	92.8	75.76	91.76	88.5
RobustLR	-	81.84	94.12	75.48	93.76	88.5
DISC	73.72	80.28	92.28	77.44	92.28	87.1
DNLL	75.4	81.91	94.24	77.9	94.21	89.2

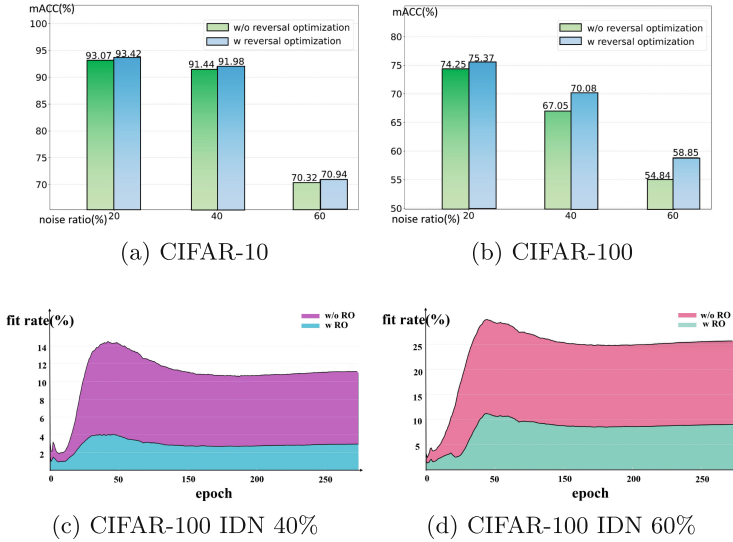


Fig. 5. (a) and (b) represent the comparison of the mACC metrics between the usage and non-usage of the enhanced-selection and reversal optimization module, and (c), (d) represent the comparison of the noise fit rates between the usage and non-usage of the enhanced-selection and reversal optimization module.

4.3 Ablation Study

The Enhanced-Selection strategy relies on noise probabilities derived from noise estimation with implicit regularization. Without the implicit regularizer, this strategy reverts to the classical Gaussian Mixture Model (GMM)-fitted small-loss selection approach. Within the DNLL framework, we aim to exclude either the implicit regularizer or the reversal optimization module to evaluate the individual contributions and interactions of each component. Experimental results on the CIFAR-10 and CIFAR-100 datasets—both containing high proportions of instance-dependent noise—are presented in Table 3. The results indicate that removing the implicit regularizer significantly degrades performance. This decline stems not only from unaddressed confirmation bias during the model learning process but also from inaccuracies in the clean set, which render reversal optimization ineffective. Although removing the reversal optimization results in a slight performance decline, this decline becomes more pronounced with increasing dataset imbalance. Overall, DNLL demonstrates superior performance, confirming that considering model biases under label noise contamination is crucial and illustrating the synergistic integration of the three proposed modules.

In order to gain a better understanding of the methodology used in this study, we also analyzed the tricks employed in Table 4, which include Mixup augmentation, strong augmentation, and co-training. Among these techniques, co-training contributes the least, followed by Mixup, and then strong augmentation. Removing strong augmentation would prevent the optimization of con-

Table 3. Ablation study of the proposed modules without any trick. IR represents the implicit regularizer and RO represents the reversal optimization module.

Moudules		CIFAR-10		CIFAR-100	
IR	RO	Inst.40	Inst.60	Inst.40	Inst.60
		95	50.37	62.55	41.25
✓		95.18	90.1	73.58	62.63
	✓	95.2	50.92	62.99	41.51
✓	✓	95.64	90.63	76.46	65.98

Table 4. Ablation study of tricks.

DataSet	CIFAR-10		CIFAR-100	
Noise Type	Inst.40	Inst.60	Inst.40	Inst.60
DNLL	96.59	96.05	81.33	75.54
w/o Mixup	95.23	90.76	76.17	70.45
w/o strong aug.	95.37	87.42	77.42	63.42
w/o co-train	96.31	95.70	80.04	73.60

sistent representations between strong and weak augmentations, while removing Mixup would hinder the achievement of consistent representations for local features. Strong augmentation and Mixup serve as auxiliary and facilitative roles in optimizing the reversal optimization module.

4.4 Discussion of Debiasing Effect and Scalability

Research on ablation has demonstrated that the proposed modules significantly enhance testing accuracy, thereby indicating a reduction in confirmation bias. To further validate the effectiveness of the proposed method on mitigating model biases, we analyzed the mean accuracy (mACC) with and without the enhanced selection and reversal optimization under varying ratios of noisy labels, as depicted in Figure 5a and Figure 5b.

The mean Accuracy metric (mACC) effectively highlights fairness, particularly in datasets with multiple classes or high levels of noise. In such contexts, class imbalance is often more pronounced. Without employing the reverse optimization module, mACC results significantly decline, indicating the importance of the proposed modules in mitigating biases. Reducing biases consequently enhances overall accuracy. For instance, in the CIFAR-100 dataset with 60% Instance-Dependent Noise (IDN), an increase in mACC from 54.84% to 58.85% corresponds to an improvement in Accuracy (ACC) from 62.63% to 65.98%.

To further elucidate our findings, we plot the noise fitting ratio over time for CIFAR-100 datasets with IDN 40% and IDN 60%, as shown in Figure 5c and Figure 5d, respectively. The noise fitting ratio is defined as the proportion

of predictions that match the noisy labels but do not match the ground truth. Utilizing our modules results in a notable reduction in noise fitting ratios.

Compared to other methods that require extensive hyperparameter tuning, our approach necessitates the adjustment of significantly fewer hyperparameters. Specifically, only the learning rate parameters need to be fine-tuned. This streamlined requirement significantly enhances the scalability of our method.

5 CONCLUSION AND DISCUSSION

This paper is the first to propose a novel approach that integrates sample selection methods under the context of label noise contamination, while simultaneously ensuring both fairness and robustness. We introduce a unified framework that aims to address these issues through a core mechanism of defining worst-case decision boundaries and conducting implicit compression via adversarial optimization. Experimental results validate the efficacy of our framework in achieving these objectives. However, our approach has certain limitations. Specifically, while we have discussed inter-class fairness, intra-class fairness remains an area that requires further exploration. In future work, we aim to extend our methodology to mitigate biases inherent in pre-trained models, thereby enhancing overall fairness and robustness.

References

1. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems* **32** (2019)
2. Chen, M., Cheng, H., Du, Y., Xu, M., Jiang, W., Wang, C.: Two wrongs don't make a right: Combating confirmation bias in learning with label noise. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 37, pp. 14765–14773 (2023)
3. Chen, P., Liao, B.B., Chen, G., Zhang, S.: Understanding and utilizing deep neural networks trained with noisy labels. In: *International Conference on Machine Learning*. pp. 1062–1070. PMLR (2019)
4. Chen, W., Zhu, C., Chen, Y., Li, M., Huang, T.: Sample prior guided robust model learning to suppress noisy labels. *arXiv preprint [arXiv:2112.01197](https://arxiv.org/abs/2112.01197)* (2021)
5. Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., Liu, Y.: Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint [arXiv:2010.02347](https://arxiv.org/abs/2010.02347)* (2020)
6. Feng, C., Tzimiropoulos, G., Patras, I.: Ssr: An efficient and robust framework for learning with unknown label noise. *arXiv preprint [arXiv:2111.11288](https://arxiv.org/abs/2111.11288)* (2021)
7. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* **31** (2018)
8. 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)

9. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. pp. 630–645. Springer (2016)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
11. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint [arXiv:2002.07394](https://arxiv.org/abs/2002.07394) (2020)
12. Li, W., Wang, L., Li, W., Agustsson, E., Van Gool, L.: Webvision database: Visual learning and understanding from web data. arXiv preprint [arXiv:1708.02862](https://arxiv.org/abs/1708.02862) (2017)
13. Li, Y., Han, H., Shan, S., Chen, X.: Disc: Learning from noisy labels via dynamic instance-specific selection and correction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 24070–24079 (2023)
14. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. *Adv. Neural. Inf. Process. Syst.* **33**, 20331–20342 (2020)
15. Liu, S., Zhu, Z., Qu, Q., You, C.: Robust training under label noise by over-parameterization. In: *International Conference on Machine Learning*. pp. 14153–14172. PMLR (2022)
16. Liu, S., Zhu, Z., Qu, Q., You, C.: Robust training under label noise by over-parameterization. In: *International Conference on Machine Learning*. pp. 14153–14172. PMLR (2022)
17. Liu, X., Shen, F., Zhao, J., Nie, C.: Randommix: A mixed sample data augmentation method with multiple mixed modes. arXiv preprint [arXiv:2205.08728](https://arxiv.org/abs/2205.08728) (2022)
18. Liu, Y., Guo, H.: Peer loss functions: Learning from noisy labels without knowing noise rates. In: *International conference on machine learning*. pp. 6226–6236. PMLR (2020)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
20. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural. Inf. Process. Syst.* **33**, 596–608 (2020)
21. Song, H., Kim, M., Lee, J.G.: Selfie: Refurbishing unclean samples for robust deep learning. In: *International Conference on Machine Learning*. pp. 5907–5915. PMLR (2019)
22. Song, H., Kim, M., Lee, J.G.: Selfie: Refurbishing unclean samples for robust deep learning. In: *International Conference on Machine Learning*. pp. 5907–5915. PMLR (2019)
23. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022)
24. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 31 (2017)
25. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: A joint training method with co-regularization. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 13726–13735 (2020)
26. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2691–2699 (2015)

27. Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. *Adv. Neural. Inf. Process. Syst.* **33**, 6256–6268 (2020)
28. Zhang, Y., Zheng, S., Wu, P., Goswami, M., Chen, C.: Learning with feature-dependent label noise: A progressive approach. arXiv preprint [arXiv:2103.07756](https://arxiv.org/abs/2103.07756) (2021)
29. Zhao, G., Li, G., Qin, Y., Liu, F., Yu, Y.: Centrality and consistency: two-stage clean samples identification for learning with instance-dependent noisy labels. In: *European Conference on Computer Vision*. pp. 21–37. Springer (2022)
30. Zhao, P., Yang, Y., He, Q.C.: Implicit regularization via hadamard product over-parametrization in high-dimensional linear regression. arXiv preprint [arXiv:1903.09367](https://arxiv.org/abs/1903.09367) **2**(4), 8 (2019)



SEDA: Similarity-Enhanced Data Augmentation for Imbalanced Learning

Javad Sheikh¹(✉) , Farshad Farahnakian¹ , Fahimeh Farahnakian^{1,2} ,
Luca Zelioli¹ , and Jukka Heikkonen¹

¹ Department of Computing, University of Turku, 20500 Turku, Finland
{javshe, farfar, fahfar, luzeli, jukhei}@utu.fi

² Geological Survey of Finland (GTK), 02151 Espoo, Finland

Abstract. Imbalanced datasets can significantly affect the performance of Machine Learning (ML) models, as they tend to overfit to the majority class and struggle to generalize well for minority classes. To mitigate these issues, we introduce an augmentation technique called Similarity-Enhanced Data Augmentation (SEDA) for handling imbalanced datasets. SEDA integrates feature and distance similarities to augment the minority samples. By incorporating feature importance, SEDA ensures that the most influential features are prioritized, leading to more meaningful synthetic samples. We evaluated the impact of SEDA on the performance of four ML models, including Multi-Layer Perceptron (MLP), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR). SEDA's effectiveness is compared against random and SMOTE oversampling methods. Experimental results are collected on geophysical data from Lapland, Finland. The dataset exhibits a significant class imbalance, comprising 15 known samples in contrast to 2.92×10^5 unknown samples. Experiments show that adding high-quality synthetic samples can help the model to generalize better to unseen data, addressing the overfitting issue commonly seen in imbalanced datasets. A part of the implemented methodology of this work is integrated in QGIS as a new toolkit which is called EIS Toolkit (https://github.com/GispoCoding/eis_toolkit) for mineral prospectivity mapping.

Keywords: imbalanced learning · data augmentation · feature importance · machine learning methods · distance similarity measurement

1 Introduction

The ever-expanding volume of data in complex systems and the remarkable progress in machine learning (ML) techniques have empowered artificial intelligence (AI)-based systems to carry out tasks such as classification and detection traditionally performed by humans. Furthermore, the significant amount of data exceeds the human capacity for analysis, therefore we require to automatically perform the tasks by employing ML models trained for different applications. Automating classification across various sectors including health [2],

security [25], mineral [10], and agriculture [11] promises to enable organizations and companies to operate with greater efficiency and impact. The classification algorithms require high-quality training datasets, whereby a set of data points or observations are labeled for analysis and processing by ML models to extract and capture the patterns between the features of the data points and their ground truths (labels). However, collecting label data or ground truth is not an easy task in many real-world applications, and workers should spend a lot of time annotating and labeling data. As a result, ML engineers and data scientists have to continually deal with imbalanced datasets in real-world applications [14].

Imbalanced datasets pose significant challenges in classification tasks: (1) primarily through bias in prediction, (2) misleading accuracy metrics, and (3) overfitting to the majority class [4]. There are various methods for addressing imbalanced data challenges and improving the performance of ML models on train data for classification tasks, each with advantages and limitations. Despite significant progress in addressing imbalanced learning, no approaches have taken into account both feature and distance similarities to generate high-quality synthetic samples. SEDA aims to balance the training data and enhance the generalization of the ML model by prioritizing the most influential features. This paper aims to answer the following research question: "How can SEDA enhance the performance of ML models by augmenting minority known samples from unknown samples?". The main contributions of this work are:

1. Integration of feature similarities: SEDA leverages feature similarity measures, including Principal Component Analysis (PCA) [19], Independent Component Analysis (ICA) [18] and entropy-based [8] methods, to assess the significance of each feature. This ensures that critical features influencing the model's performance are accurately identified and utilized.
2. Distance-based augmentation: By calculating distance similarities (e.g., Euclidean, Manhattan, and Cosine) between samples and incorporating weighted feature importance, SEDA ensures that the augmented samples maintain both the statistical integrity and the relevance of the original data distribution.
3. Algorithmic level imbalanced data handling: At the algorithmic level, we adjust the decision threshold of a model to balance the trade-off between false positives and false negatives.
4. Grid Search: We utilized grid search to identify the optimal values for the oversampling rate, decision threshold, and model hyperparameters.
5. Comprehensive evaluation: We also provide an extensive evaluation of our method with four common ML models, including Multi-Layer Perceptron (MLP), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR). In addition, SEDA is compared with random oversampling, ADASYN, SMOTE, and Borderline-SMOTE .
6. Real-world highly imbalanced dataset: To evaluate the proposed framework, we used a highly unbalanced data set related to mineral deposits found in Kolari municipalities in Finland. Imbalanced class issues are commonly encountered in mineral resource exploration projects, as creating a well-

balanced dataset for classification and generating prospectivity maps for mineral deposits can be quite costly. The study offers valuable framework and insights that can help decision makers and mineral exploration companies in planning, identifying areas with known mineralization, and discovering new exploration targets, while also saving money and energy.

The remainder of this paper is organized as follows. Section 2 reviews the most closely related work dealing with imbalanced datasets. The proposed method for handling imbalanced data is described in Section 3. Sections 4 and 5 provides the experimental design and results, respectively. The discussion and conclusions are drawn in Section 6.

2 Related Work

The techniques proposed for handling imbalanced datasets can be categorized into four groups: (1) data-level approaches (2) algorithm-level approaches (3) hybrid approaches and (4) data augmentation and generation. In the following paragraphs, we briefly review each group to understand how techniques can handle imbalanced data. Additionally, a summary of previous studies conducted on this problem can be found in Table 1.

Table 1. Summary of Existing Works on Handling Imbalanced Datasets.

Technique	Application Domain	Year	Citation
SMOTE	Fraud Detection	2002	[5]
Undersampling	Spam Detection	2009	[25]
SMOTE + Tomek Links	Bioinformatics	2004	[1]
Cost-Sensitive SVM	Medical Diagnosis	1999	[31]
Balanced Random Forest	Credit Scoring	2004	[6]
One-Class SVM	Network Intrusion Detection	2001	[29]
GANs	Rare Disease Data Augmentation	2018	[13]
Isolation Forests	Cybersecurity	2008	[23]
Deep Learning (Class Weighting)	Image Recognition	2020	[22]
Transfer Learning	Medical Imaging	2018	[7]

Data-Level Approaches: Various techniques focus on adjusting the training data to balance class distributions. Three common methods include over-sampling, undersampling, and the combination of over and under-sampling. In oversampling techniques like SMOTE (Synthetic Minority Over-sampling Technique), borderline SMOTE, and support vector machine SMOTE, synthetic examples of the minority class are created by interpolating between existing minority instances [5]. In the undersampling technique, data points are randomly removed from the majority class to balance the dataset, although this

can risk losing important information. There are also some popular undersampling techniques, like random undersampling, repetitive undersampling based on ensemble models, and Tomek’s link undersampling [21]. Lastly, the combination of oversampling and undersampling methods with data cleaning techniques, like SMOTE + Tomek Links. In this method, SMOTE generates synthetic samples and Tomek Links cleans noisy samples by removing overlapping examples between classes [21].

Algorithm-Level Approaches: These methods involve modifying the learning algorithms to be more sensitive to the minority class. One of the noted algorithm-based methods that target the problem of imbalanced learning is the cost-sensitive learning function, which assigns a higher misclassification cost to the minority class [27]. Also, ensemble methods such as bagging, boosting, and stacking can enhance the model performance on imbalanced datasets. Combining various model and assigning higher weights can provide more emphasis on underrepresented classes, thereby improving their predictive capability [28].

Hybrid Approaches: Combining both data-level and algorithm-level strategies, hybrid approaches aim to leverage the benefits of both. Techniques such as SMOTE followed by a cost-sensitive learning algorithm are employed to simultaneously balance the data and adjust the algorithm’s, focus on minority classes, leading to improved performance. In [24], they developed weak classifiers using the Support Vector Machine model, and assigned two distinct misclassification cost values for each of the two classes. Then they combined the weak classifiers with undersampling and bagging techniques to create the final strong classifier.

Data Augmentation and Generation: These techniques involve generating new synthetic data to enhance the representation of the minority class. Methods like Generative Adversarial Networks (GANs) are capable of creating highly realistic synthetic samples, augmenting the minority class, and providing more data points for training, which assists in decreasing the model’s bias over the majority class [9]. Generally, GANs-based models consist of two parts: the generator and the discriminator. The generator is a convolutional neural network and the discriminator is a deconvolutional neural network [9]. In [3], a GAN-based technique for creating synthetic data to train a fraud detection classifier was introduced. The proposed model has shown acceptable results in addressing the challenge of class imbalance for a real-world gambling fraud dataset, outperforming traditional oversampling and undersampling methods.

3 Similarity-based Minority Augmentation (SEDA) Technique

Fig.1 depicts the overall SEDA framework. To handle an imbalanced dataset, SEDA evaluates and ranks the importance of each feature in terms of distance and feature similarity. It means the created balanced dataset is obtained by oversampling minority samples from the unlabeled data based on feature similarities in order to improve the performance of ML models. SEDA first utilizes feature

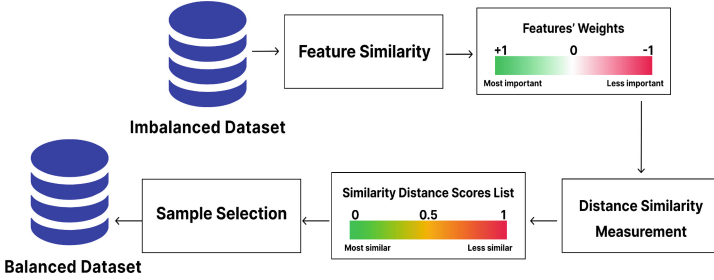


Fig. 1. The framework of SEDA technique.

similarity measures to assign weights to features, with values ranging from 0 to 1. These weights indicate the importance of each feature in the dataset. The weights w_i are computed based on the following methods:

1. Mutual Information (Entropy-Based) [8]: measures the amount of information or uncertainty associated with each feature by

$$H(x) = - \sum p(x) \log p(x) \quad (1)$$

where $p(x)$ is the probability distribution of the feature values. The entropy values are then normalized to a range of [0, 1] by applying Min-Max normalization:

$$w_i = \frac{H(X_i) - \min(H)}{\max(H) - \min(H)} \quad (2)$$

2. Principal Component Analysis (PCA) [19]: identifies the most important features based on the variance they capture. We first perform PCA on the dataset to obtain the principal components and their corresponding eigenvalues. Then, the eigenvalues are normalized to a range [0, 1] to represent the weights of the features:

$$w_i = \frac{\lambda_i - \min(\lambda)}{\max(\lambda) - \min(\lambda)} \quad (3)$$

where λ_i is the eigenvalue corresponding to the i -th principal component.

3. Independent Component Analysis (ICA) [18]: finds components that are statistically independent from each other. After performing ICA, we calculate the variance of each independent component. The variances are normalized to a [0, 1] range to assign feature weights:

$$w_i = \frac{\sigma_i^2 - \min(\sigma^2)}{\max(\sigma^2) - \min(\sigma^2)} \quad (4)$$

where σ_i^2 is the variance of the i -th independent component.

In each of these methods, the resulting weights for each feature are scaled between 0 and 1, where 0 indicates no importance and 1 indicates maximum importance. Then, SEDA calculates the distance similarity between features, and these weights are then used in the weighted distance similarity calculations. Distance similarity measures are:

1. Euclidean distance: measures the straight-line distance between points.
2. Manhattan distance: measures the sum of absolute differences
3. Cosine similarity: measures the cosine of the angle between two vectors

SEDA ranks the majority of unknown samples based on their scores for each minority sample and chooses the top-ranked samples to augment synthetic minority samples and balance the dataset. The sample selection process involves evaluating various oversampled rates and selecting the optimal number that enhances the performance of ML models. This dual approach ensures that the method accounts for the most critical features while accurately measuring the distances between samples. The pseudocode 1 describes the steps involved in the SEDA method to generate synthetic samples.

Algorithm 1. SEDA: Similarity-Enhanced Data Augmentation

Input: Imbalanced dataset $\mathcal{D}_{\text{imbalanced}}$ with majority and minority classes; Number of synthetic samples to generate N ; Distance similarity measure $\mathcal{D}_{\text{similarity}}$ (e.g., cosine, Euclidean)

Output: Balanced dataset with synthetic samples $\mathcal{D}_{\text{balanced}}$

- 1: **Step 1: Compute Feature Similarity (e.g., Entropy, PCA, ICA)**
- 2: Assign weights $w_i \in [0, 1]$ to each feature i based on their importance scores.
- 3: **Step 2: Compute Weighted Distance Similarity (e.g., Euclidean, Manhattan, Cosine)**
- 4: **for** each minority sample x_{min}^k **do**
- 5: **for** each majority sample x_{maj}^j **do**
- 6: Compute weighted distance d_{kj} using the chosen distance similarity measure:

$$d_{kj} = \sum_{i=1}^m w_i \cdot \mathcal{D}_{\text{similarity}}(x_{\text{min}}^{k,i}, x_{\text{maj}}^{j,i})$$

where w_i is the weight of feature i .

- 7: **end for**
 - 8: Calculate a score for each majority sample based on d_{kj} .
 - 9: Rank majority samples based on their scores.
 - 10: **end for**
 - 11: **Step 3: Select Synthetic Samples**
 - 12: **for** each minority sample x_{min}^k **do**
 - 13: Select N top-ranked majority samples $\{x_{\text{maj}}^{j_1}, x_{\text{maj}}^{j_2}, \dots, x_{\text{maj}}^{j_N}\}$ from Step 2 as candidates.
 - 14: **end for**
 - 15: **Step 4: Balance the Dataset**
 - 16: Remove $N \times$ number of minority samples from the majority class and add them to the minority class.
 - 17: **return** $\mathcal{D}_{\text{balanced}}$
-

4 Experiments

Fig.2 illustrates the proposed framework for conducting our experiments. Initially, the original data set is divided into training and test sets, as shown in Figure 1. Next, SEDA is applied to the training set to generate a balanced dataset. Subsequently, an ML model is trained on this balanced dataset using 6-fold Stratified cross-validation (SCV). During this process, grid search is

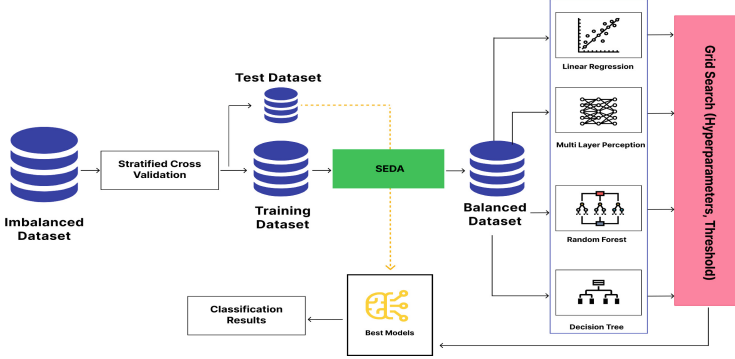


Fig. 2. The flowchart for experiment stage.

employed to find the best values for the oversampling rate, the decision threshold, and the hyperparameters. Finally, the best-performing model is evaluated on the test dataset to produce classification results.

To find the best model and evaluate its performance in cross-validation (CV), we employed the geometric mean (G_{mean}) as a loss function. Unlike traditional classification loss functions that focus primarily on minimizing misclassifications, G_{mean} considers the distance between sample features. By accounting for both sensitivity (recall) and specificity, G_{mean} provides a balanced evaluation metric that is robust to class imbalance [16].

$$G_{mean} = \sqrt{Sensitivity \times Specificity} \tag{5}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{7}$$

where TP , FP , TN and FN indicate the total number of true positive, false positive, true negative, and false negative pixels, respectively.

4.1 Data

We evaluated our proposed framework on a very highly imbalanced geophysical dataset for the Mineral Prospectivity Mapping (MPM) application. MPM aims to predict the likelihood of finding specific types of mineral deposits for mineral exploration, and resource assessment. The data contains 15 deposit samples versus 2.92×10^5 unknown samples. Our study area is located in the municipalities of Kittilä, Kolari, and Muonio, Lapland, Finland (Figure 3). We have three types of input data, which collectively generate 13 different attributes for each sample.

- **Airborne Electromagnetic (AEM)** includes measurements of the electrical conductivity of the earth’s subsurface¹. This data contains 4 features.
- **Magnetic data** used to identify variations in the Earth’s magnetic field caused by the magnetic properties of subsurface rocks². Totally, we have 5 features from this data.
- **Radiometric data** gives the measurement of natural gamma radiation emitted from the earth’s surface to infer the concentration of radio element isotopes such as uranium (U), thorium (Th), and potassium (K)³. This data contains 4 features.

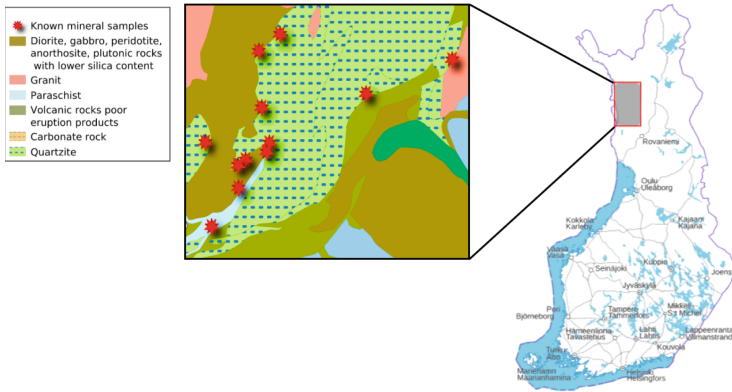


Fig. 3. Study area.

4.2 Machine learning algorithms

We used four different ML algorithms for our experiments as follows:

1. Multi-Layer Perceptron (MLP) [26] is a type of artificial neural network characterized by its layered structure. One of the principal attributes of MLPs is universal approximation capability which means they have the theoretical ability to approximate any continuous function to a desired degree of accuracy, given sufficient neurons in the hidden layers.
2. Random Forest (RF) [20] is capable of processing and analyzing large datasets that contain a high number of features or variables.

¹ https://tupa.gtk.fi/paikkatieto/meta/aeroelectromagnetic_raster_data_of_finland.html

² https://tupa.gtk.fi/paikkatieto/meta/aeromagnetic_raster_data_of_finland.html

³ https://tupa.gtk.fi/paikkatieto/meta/aeroradiometric_raster_data_of_finland.html

3. Decision Tree (DT) [30] is a non-linear predictive model which is structured like a tree. It can effectively handle large datasets with a high number of features. One of the main advantages of DTs is their interpretability and simplicity.
4. Logistic Regression (LR) [12] uses a logistic function, a special S-shaped curve that transforms any input into a value between 0 and 1 (probability) representing probabilities for classification purposes.

Table 2 lists the key ML parameters and their best values. The parameter names at the table align with the standard parameter names used in the Scikit-learn Python package. The optimal values for these parameters for each model were obtained through a grid search during cross-validation.

Table 2. Main hyper-parameter of ML models and their best value.

ML Model	Hyper-parameter	Search space	Optimal Hyper-parameter Value
LR	penalty	l2, l1 ,elasticnet, None	l2
MLP	alpha	0.0001, 0.001, 0.01	0.0001
	hidden_layer_sizes	[(2), (4), (8) , (2,4), (4,8), (2, 8)]	(2)
DT	criterion	gini, entropy, log_loss	gini
	splitter	best, random	best
	max_depth	2, 8, None	8
RF	n_estimators	4, 8, 16	16
	max_depth	4, 8, 16, None	None
	criterion	gini, entropy	gini

5 Results

5.1 Feature and distance similarity impact on ML performance

In this subsection, we answer this question "Which feature similarity method and distance measure combination yielded the best overall performance?". Fig.4 (a) shows the performance (measured by G_{mean}) of the four ML models using different feature similarity methods (e.g., PCA, ICA, Entropy, and None (without feature importance)). The results show that PCA consistently provides the highest performance across LR, MLP, and DT models, indicating its effectiveness in capturing the most relevant features. Entropy-based similarity shows the highest performance for DT.

Fig.4 (b) depicts the performance (G_{mean}) of the four ML models using different distance measures (e.g., Euclidean, Manhattan, Cosine). For three LR, RF, and MLP models, all distance metrics show similar results. However, Cosine distance measure consistently yields the best result for DT. Combining insights from both plots, the combination of PCA for feature similarity and Cosine distance measure yields the best overall performance for all four machine learning models.

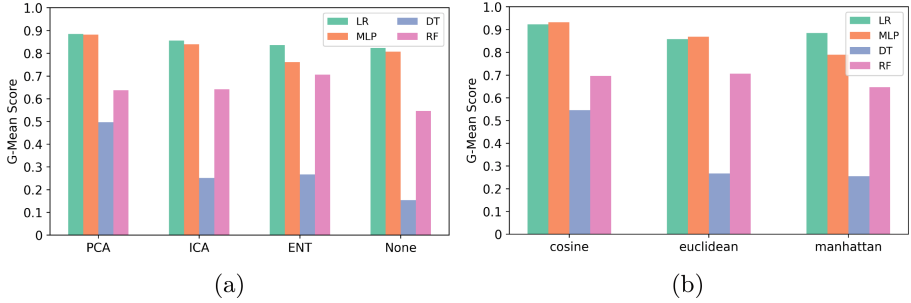


Fig. 4. Performance comparison of ML models using different (a) feature similarity methods and (b) distance similarity.

5.2 Impact of augmentation rate on ML performance

We collected the results to answer this question "How does augmentation rate (N in Algorithm 1) impact the performance of each ML model?". $N=n$ indicates the $n\%$ percentage of the top-ranked unknown samples are added to the known samples. For example, $N=50$ means that 50% of the top-ranked unknown samples are added to the known samples to balance the imbalanced dataset.

Based on the findings presented in Fig.4, this subsection summarizes the results focusing on cosine distance with PCA and entropy measures.

Table 3 shows the G-means performance of four ML models (MLP, RF, DT, LR) across varying augmentation rates N applied to the majority class. The results illustrate how different values of N influence the ability of each model to handle imbalanced datasets. Notably, $N=0.005$ (12/250,000) represents training on the original imbalanced dataset, with the augmentation process halting due to the observed elimination of sample discrimination at higher rates.

The overall results demonstrate that SEDA can enhance ML performance by generating balanced datasets from initially imbalanced ones ($N=0.005$). Specifically, LR achieved the highest G-means of 85.5% with PCA when $N=0.1$, while DT achieved 49.6%. This indicates that a moderate amount of synthetic data helps the LR and DT models perform better. MLP outperforms other models with 88.2% accuracy with PCA and a relatively small augmentation rate $N=0.025$, and RF achieved 64.3% using the entropy measure at $N=50$. RF showed improvement with higher augmentation rates, indicating its robustness to large synthetic datasets.

5.3 Comparison of over-sampling methods

SEDA is compared with four baseline methods, including:

1. Random oversampling: instead of the top-ranked samples, we randomly select N unknown samples and increase the minority samples. The value of N is selected based on Table 3 where the ML models have the best performance.

Table 3. Impact of number of selected sample rate (N) on the performance (G_{mean}) of ML models on the test dataset.

N	0.005		0.025		0.1		0.5		1.5		5		20		50	
Importance	PCA	ENT	PCA	ENT	PCA	ENT	PCA	ENT	PCA	ENT	PCA	ENT	PCA	ENT	PCA	ENT
LR	65.9	56.2	84.1	80.1	85.5	82.1	82.3	81.1	80.0	80.8	80.7	80.6	71.3	75.5	61.9	62.3
MLP	80.1	37.0	88.2	47.2	27.3	75.1	70.7	65.2	52.1	57.1	65.6	66.7	65.9	67.9	61.9	62.2
DT	0.0	0.0	46.0	22.7	49.6	9.4	15.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RF	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	9.6	45.6	34.1	58.1	54.5	54.1	64.3

2. Synthetic Minority Oversampling Technique (SMOTE) [5] first identifies k nearest neighbors for each minority class data point. Then, it randomly selects one of these neighbors and generates a new synthetic data point along the line segment connecting the minority class data point and its randomly selected neighbor.
3. Borderline-SMOTE [15] focuses on samples from the minority class close to the decision boundary. It detects minority samples most susceptible to misclassification and generates new points specifically in these areas using SMOTE.
4. Adaptive Synthetic sampling (ADASYN) [17] is prioritizes the creation of synthetic data points near the decision boundary, which are the data points that are most difficult for the classifier to classify correctly. This is done by assigning higher weights to minority class data points that are closer to the decision boundary.

As SMOTE, Borderline-SMOTE and ADASYN generate new synthetic samples from the minority class, we randomly reduced the majority class by the same number to maintain consistency with the N values used for random oversampling and SEDA. Fig.5 illustrates the ROC curves, plotting the false positive rate against the true positive rate for four distinct classifiers. These curves are pivotal for assessing the Area Under the Curve (AUC) across all models compared to the baseline methods. A higher AUC indicates superior classification accuracy for the respective algorithms.

In the ROC curve analysis for the LR model (Fig.5(a)), demonstrates that the SEDA algorithm outperforms ADASYN, SMOTE, and BorderlineSMOTE with marginally higher true positive rates at higher false positive rates, closely rivaling these methods. In the case of the MLP model (Fig.5(b)), the SEDA algorithm consistently surpasses Borderline-SMOTE and random oversampling methods across different thresholds, as evidenced by its higher AUC values. Its performance is comparable to that of ADASYN and SMOTE, showing similar effectiveness. The ROC curves for the DT model (Fig.5(c)) reveal that the SEDA algorithm achieves a significantly better true positive rate at all levels of false positive rates compared to SMOTE, Borderline-SMOTE, and ADASYN, and far outperforms the Random method in effectively classifying imbalanced datasets. For the RF model (Fig.5(d)), SEDA slightly outperforms ADASYN in

handling imbalanced datasets, particularly at higher false positive rates, with both algorithms significantly outperforming SMOTE, Borderline-SMOTE, and the Random method.

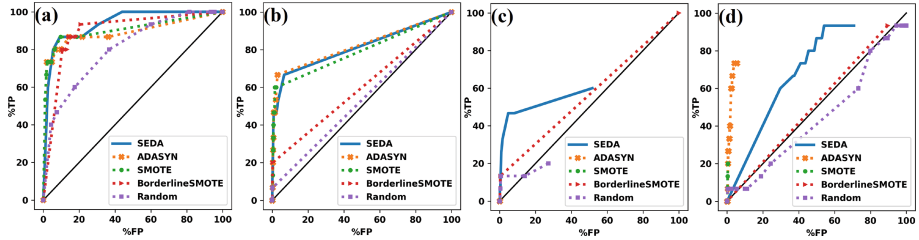


Fig. 5. Comparison of SEDA, ADASYN, SMOTE, Borderline-SMOTE and random over-sampling for (a) LR, (b) MLP, (c) DT, and (d) RF.

5.4 Decision Thresholds

Thresholding allows the model to fine-tune the classifier’s decision boundary to better account for class imbalance. Fig.6 shows the sensitivity-specificity curve and illustrates the adjustment of the decision threshold based on the trade-off between sensitivity and specificity. Each color represents a different model, with solid lines indicating sensitivity and dashed lines indicating specificity at various thresholds. The MLP model achieves high sensitivity without a significant loss in specificity, highlighting its robustness.

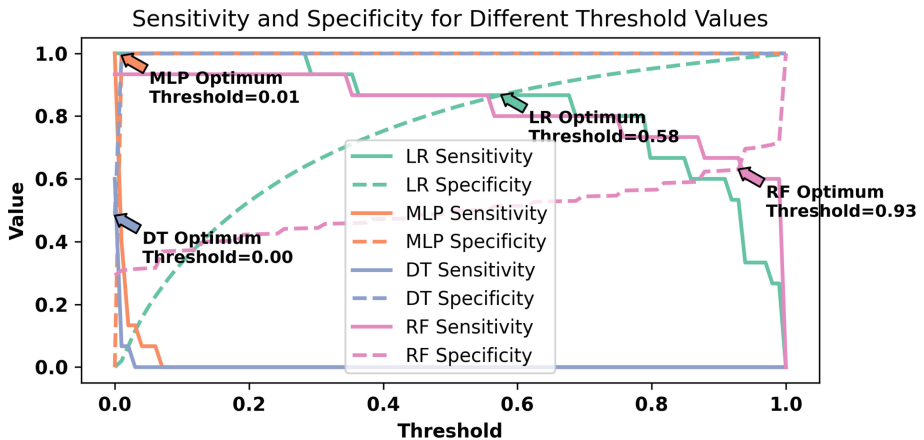


Fig. 6. Sensitivity-specificity curves of ML models based on different thresholds.

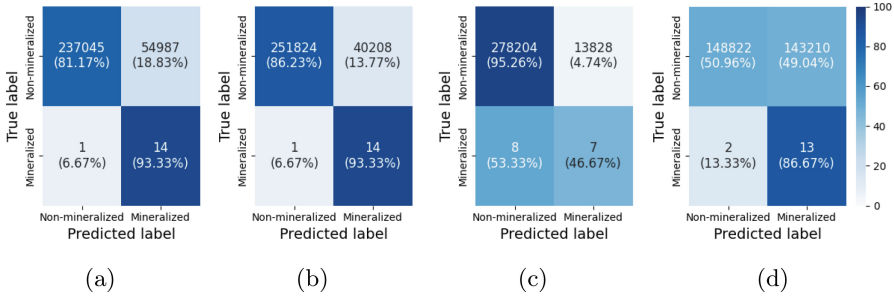


Fig. 7. Confusion matrix of (a) LR, (b) MLP, (c) DT, and (d) RF.

5.5 Confusion Matrix

The confusion matrix results on the test dataset which are depicted in Fig.7 shows that MLP and LR can get the maximum correct observations belongs to class “mineralized”. For DT, as illustrated in Fig.7, the highest correct observations belong to the classes “non-mineralized” which is 95.26%.

5.6 Prediction maps

Fig.8 shows the mineral prospectivity maps produced by the four models. To generate these maps, we used the entire dataset to train the models, utilizing the optimal values for the model’s hyperparameters, decision threshold, and the number of oversampling and undersampling instances.

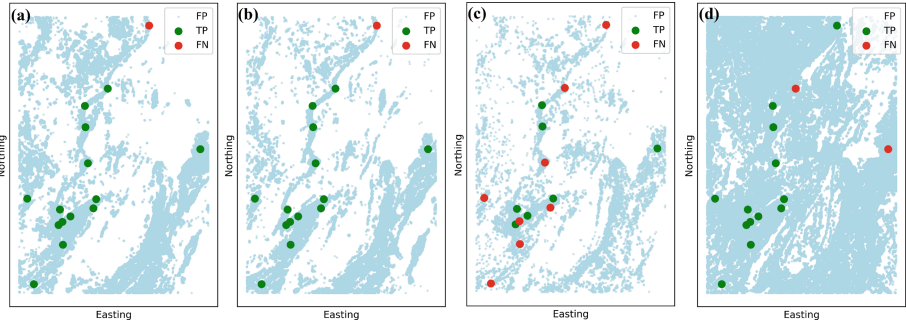


Fig. 8. Prospectivity mineral maps for (a) LR, (b) MLP, (c) DT, and (d) RF models. The known samples have been visually exaggerated to make them visible.

6 Conclusion

This paper presents a novel data augmentation method, SEDA, for handling imbalanced datasets in order to improve the performance of ML algorithms.

SEDA ensures that the generated synthetic samples are more relevant and informative by prioritizing more important features. Considering both feature importance and distance similarity helps in creating synthetic samples that improve the model’s ability to generalize to unseen data. We evaluated SEDA on a highly imbalanced dataset, where there is a significant disparity between the number of samples in the minority and majority classes in the training data. The results demonstrate that SEDA outperforms existing imbalanced data handling methods on a real dataset. SEDA can be applied to various types of datasets and is particularly effective for those with high dimensionality and complex feature interactions. In the future, we plan to test the SEDA technique on imbalanced datasets collected for different applications. This comprehensive approach highlights SEDA’s contributions in addressing imbalanced datasets, improving ML model performance, and facilitating practical applications in geospatial analysis through the EIS Toolkit integration.

Acknowledgements. The compilation of the presented work is supported by funds from the Horizon Europe research and innovation program under Grant Agreement number 101057357, EIS - Exploration Information System. For further information, check the website: [EIS](#).

References

1. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.* **6**(1), 20-29 (jun 2004). <https://doi.org/10.1145/1007730.1007735>
2. Butt, U.M., Letchmunan, S., Ali, M., Hassan, F.H., Baqir, A., Sherazi, H.H.R.: Machine learning based diabetes classification and prediction for healthcare applications. *Journal of healthcare engineering* **2021**(1), 9930985 (2021)
3. Charitou, C., Dragicevic, S., d’Avila Garcez, A.: Synthetic data generation for fraud detection using gans (2021)
4. Chawla, N.V.: Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook* pp. 875–886 (2010)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002)
6. Chen, C., Breiman, L.: Using random forest to learn imbalanced data. University of California, Berkeley (01 2004)
7. Cheplygina, V., de Bruijne, M., Pluim, J.P.W.: Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis (2018)
8. Cover, T.M., Thomas, J.A.: *Elements of Information Theory* 2nd Edition (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience (July 2006)
9. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **35**(1), 53–65 (2018)
10. Farahnakian, F., Zelioli, L., Pitkänen, T., Pohjankukka, J., Middleton, M., Tuominen, S., Nevalainen, P., Heikkonen, J.: Multistream convolutional neural network

- fusion for pixel-wise classification of peatland. In: 2023 26th International Conference on Information Fusion (FUSION). pp. 1–8 (2023). <https://doi.org/10.23919/FUSION52260.2023.10224183>
11. Farahnakian, F., Sheikh, J., Farahnakian, F., Heikkonen, J.: A comparative study of state-of-the-art deep learning architectures for rice grain classification. *Journal of Agriculture and Food Research* **15**, 100890 (2024)
 12. Foster, D.J., Kale, S., Luo, H., Mohri, M., Sridharan, K.: *Logistic regression: The importance of being improper* (2018)
 13. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: *Synthetic data augmentation using gan for improved liver lesion classification* (2018)
 14. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. *Expert Syst. Appl.* **73**, 220–239 (2017)
 15. Han, H., Wang, W., Mao, B.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing* (2005), <https://api.semanticscholar.org/CorpusID:12126950>
 16. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics, Springer (2009), <https://books.google.fi/books?id=eBSgoAEACAAJ>
 17. He, H., Bai, Y., Garcia, E.A., Li, S.: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Conference on Neural Networks (IEEE World Congress on Computational Intelligence). pp. 1322–1328 (2008). <https://doi.org/10.1109/IJCNN.2008.4633969>
 18. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. *Neural networks : the official journal of the International Neural Network Society* **13** 4-5, 411–30 (2000), <https://api.semanticscholar.org/CorpusID:11959218>
 19. Kherif, F., Latypova, A.: Chapter 12 - principal component analysis. In: Mechelli, A., Vieira, S. (eds.) *Machine Learning*, pp. 209–225. Academic Press (2020). <https://doi.org/10.1016/B978-0-12-815739-8.00012-2>
 20. Kulkarni, V.Y., Sinha, P.K.: Pruning of random forest classifiers: A survey and future directions. In: 2012 International Conference on Data Science & Engineering (ICDSE). pp. 64–68. IEEE (2012)
 21. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **18**(17), 1–5 (2017)
 22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(2), 318–327 (2020). <https://doi.org/10.1109/TPAMI.2018.2858826>
 23. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 413–422 (2008). <https://doi.org/10.1109/ICDM.2008.17>
 24. Liu, L., Wu, X., Li, S., Li, Y., Tan, S., Bai, Y.: Solving the class imbalance problem using ensemble algorithm: application of screening for aortic dissection. *BMC Med. Inform. Decis. Mak.* **22**(1), 82 (2022)
 25. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **39**(2), 539–550 (2009). <https://doi.org/10.1109/TSMCB.2008.2007853>
 26. Popescu, M.C., Balas, V.E., Perescu-Popescu, L., Mastorakis, N.: Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems* **8**(7), 579–588 (2009)

27. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **10**(3), e0118432 (2015)
28. Salunkhe, U.R., Mali, S.N.: Classifier ensemble design for imbalanced data classification: A hybrid approach. *Procedia Computer Science* **85**, 725–732 (2016). <https://doi.org/10.1016/j.procs.2016.05.259>, international Conference on Computational Modelling and Security (CMS 2016)
29. Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, R.: Estimating support of a high-dimensional distribution. *Neural Computation* **13**, 1443–1471 (07 2001). <https://doi.org/10.1162/089976601750264965>
30. Suthaharan, S., Suthaharan, S.: Decision tree learning. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* pp. 237–269 (2016)
31. Veropoulos, K., Campbell, C., Cristianini, N.: Controlling the sensitivity of support vector machines. *Proceedings of International Joint Conference Artificial Intelligence* (06 1999)



A Robust Framework for Evaluation of Unsupervised Time-Series Anomaly Detection

Onat Gungor¹, Amanda Rios^{2(✉)}, Priyanka Mudgal², Nilesh Ahuja²,
and Tajana Rosing¹

¹ University of California, San Diego, USA
{ogungor, tajana}@ucsd.edu

² Intel Corporation, Santa Clara, USA
{amanda.rios, priyanka.mudgal, nilesh.ahuja}@intel.com

Abstract. Time-series anomaly detection (TAD) has a pivotal role across various domains ranging from manufacturing to health care monitoring. Numerous machine learning solutions have been proposed for TAD, with varying levels of complexity. However, most solutions benchmark their performance using misleading evaluation metrics which hinder reliable comparative analysis and the development of truly robust TAD methods. In the present work, we disentangle how performance evaluation can be unreliable due to several factors: suboptimal scoring functions, thresholding functions that assume access to all test labels, prediction modification based on test labels, lack of benchmarking against trivial baselines, and finally, problematic datasets. In this paper, we endeavor to address these issues by introducing a comprehensive TAD evaluation framework which includes: state-of-the-art deep-learning (DL) and traditional machine learning (ML) TAD algorithms; TAD baselines; an extensive set of scoring, thresholding and evaluation functions. Our rigorous analysis shows that: (i) TAD baselines and simple ML algorithms achieve performance often on par with advanced SOTA DL solutions. (ii) Scoring and thresholding function selection can greatly impact the anomaly prediction performance. (iii) Evaluation metrics used in the field, mostly focused on post-thresholding output, are worryingly inconsistent and can generate starkly overestimated predictions. We advocate instead for a more widespread use of pre-thresholding metrics and for post-thresholding metrics that closely correlate to the former. Our code is available at <https://github.com/intellabs/tsad-ef>.

Keywords: Time-series anomaly detection · Machine learning · Deep learning.

1 Introduction

Time series anomaly detection (TAD) seeks to identify substantial deviations from expected patterns of behavior [28]. TAD is widely used in many real-world applications

O. Gungor and A. Rios—These authors contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_4.

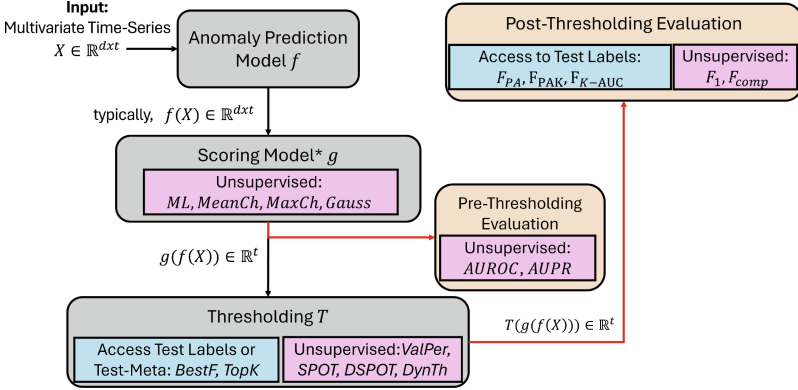


Fig. 1. Our proposed modular TAD Evaluation Framework (TADEF)

such as fault diagnosis, fraud detection, and network intrusion detection. Effectively detecting anomalies is critical for ensuring security and preventing economic losses [34]. Nevertheless, a major challenge in TAD is the lack of labeled anomalous data [6], which makes algorithms relying on balanced labeled data unsuitable. Because of this, TAD solutions often leverage Self-Supervised Learning (SSL) to identify patterns and anomalies by training on only normal data points with minimal to no supervision. Traditional self-supervised solutions to TAD span over several machine learning (ML) and statistical algorithms [4, 13, 23, 24]. More recently, in line with the success of Deep learning (DL), an avalanche of DL TAD models have been introduced with varying degrees of complexity [3, 18, 20, 33, 35].

Yet, these recent DL methods often employ unreliable evaluation paradigms, which can generate misleading performance scores and obscure the real progress achieved by scaling up model complexity over simpler solutions. Amongst the most problematic is the reliance on labeled test-time data to fine-tune decision thresholds and even model predictions [9]. For instance, in the case of the latter, an ubiquitous metric “F-PA” (point adjustment) [32, 35] considers any anomaly window with at least one correctly predicted time step as equivalent to being fully correct. As a result, the entire anomaly prediction window is modified by copying over the exact test anomalies. F-PA has been shown to make random and well-trained DL TAD model anomaly scores indistinguishable [9] and yet it continues to be used in even the most recent TAD DL methods [35]. Similarly, a frequently used thresholding function, “BestF” [26], uses all test-labels to finetune a decision threshold for TAD. But, as we will show in this paper, varying the amount of test data access or adopting unsupervised thresholding schemes can radically alter model performance, making cross-model comparisons difficult. Importantly, relying on test data labels for TAD is impractical for real-life scenarios where labeled anomaly data is often unavailable for essential tasks such as determining decision thresholds or adjusting model predictions.

Issues in TAD extend even beyond the aforementioned considerations. State-of-the-art (SOTA) DL TAD methods utilize simplistic scoring functions, which as we show

in this work, might yield suboptimal prediction performance and further hinder effective comparative TAD method analysis. For example, mean over channels, one of the most frequent scoring functions [31], collapses multidimensional error scores (from DL) into univariate scores with equal weight across channels. When dealing with complex TAD datasets, sometimes containing hundreds of sensors (dimensions), this significantly impacts performance. To alleviate this effect in high dimensional TAD datasets, we introduce a simple and lightweight ML-based scoring approach. Lastly, common TAD benchmarking datasets also have significant added complexity and/or flaws. For instance, SMAP and MSL [8] contain unlabeled anomalous training data; SWAT [7] and WADI [2] have distributional shift of normal data from train to test time [28]. These characteristics make unsupervised TAD solutions even more difficult to achieve, since most SSL TAD algorithms assume access to clean (not anomalous) training data and also, that normal data does not undergo excessive distribution shift at test-time.

To address these many challenges, we propose a rigorous TAD evaluation framework TADEF (Fig. 1) which provides diverse performance scores that, when combined, are significantly more indicative of TAD performance under real-world conditions. To this end, our main contributions are as follows:

1. To the best of our knowledge, we are the first to conduct an in-depth quantitative comparative analysis of TAD evaluation metrics, aiming to motivate future TAD research to use only robust TAD metrics instead of those with problematic assumptions. We show how TAD performance is sensitive to the thresholding function used, and hence advocate for consistent reporting of pre-thresholding metrics, e.g., AUPR. Also, based on collective cross-metric correlation scores, we advocate for the use of existing post-thresholding metrics with the highest collective correlation to their pre-thresholding predecessor such as pointwise F1-score.
2. We are releasing to the community a time-series anomaly detection evaluation framework (TADEF) comprised of a collection of modules each with diverse SOTA TAD models, datasets, baselines, scoring functions, thresholding functions, and evaluation metrics. Collectively, this modularized framework can enable more robust comparative analysis in TAD.
3. With TADEF, we generate comprehensive results that underscore the challenges and limitations of current SOTA TAD methods, scoring functions, thresholding functions and evaluation metrics in general. Overall, we show how TAD performance is often more sensitive to the choice of scoring, thresholding or evaluation function than to the choice of underlying TAD model itself.
4. Lastly, we propose more reliable variants of currently used scores, thresholding functions and metrics to further aid comparative TAD analysis.

2 Background: The State of TAD Analysis Today

2.1 Summary of existing TAD solutions

A time-ordered sequence of data points is referred to as a time-series and may contain one (univariate) or multiple real-valued (multivariate) variables (channels). Anomalies in time series can occur as individual time-points (point anomaly), a contiguous

sequence of time-points (sequence anomaly), or point/sequence across different channels (complex/contextual anomaly). In any modality, an anomaly is such that it deviates with respect to some measure, model, or embedding from the regular patterns in the data [22]. Time series anomaly detection (TAD) aims to detect and localize these deviations per time-point. Traditional TAD solutions have employed common unsupervised ML and statistical tools such as Principal Component Analysis (PCA) [24], One-Class Support Vector Machine (OCSVM) [23], Isolation Forest (IF) [13], and Local Outlier Factor (LOF) [4] as well as some sequence-specific statistical models [12,36]. With the recent explosion of DL models in other anomaly detection modalities, e.g. Language and Vision, many DL algorithms have been proposed for TAD as well. Some of those are developed exclusively for the time series modality [3,33], while others repurpose general sequence-based DL architectures such as Transformers and Autoencoders [18,20], which can easily be trained with a reconstruction-based loss function. The most recent DL solutions leverage complex self-supervised contrastive-based techniques [35,37] in an attempt to improve TAD prediction performance.

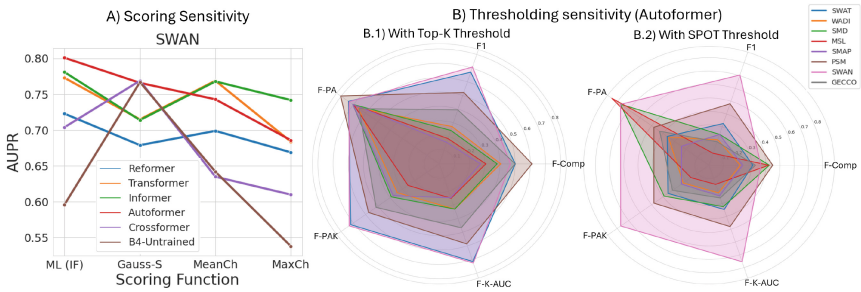


Fig. 2. Scoring and Thresholding sensitivity in DL TAD models

2.2 Limitations of SOTA TAD solutions

Anomaly detection models (time-series or otherwise) generate anomaly scores which are used to distinguish between defect-free and defective samples. The effectiveness of a model is evaluated by standard evaluation metrics such as Area Under Receiver Operating Characteristic (AUROC), Area Under Precision-Recall (AUPR), and F-metrics (F1, etc.) (please refer [5] for more details). To avoid confusion, we will use the term “score” for the anomaly score generated by the model and “metric” for quantities such as AUROC, AUPR, etc. that are used to evaluate prediction performance.

Scoring: Most TAD DL models generate multidimensional anomaly scores $S \in \mathbb{R}^{d \times t}$ as their final output during inference, with d matching input channels and t the number of time steps. To generate univariate anomaly prediction scores per time-point, these models typically employ a very simple scoring function, *MeanCh*, which consists of averaging the error scores across all output channel dimensions for a final score $S \in$

$\mathbb{R}^{1 \times t}$. While simple, this procedure may not be optimal for high multidimensional TAD datasets, as the final anomaly scores for such datasets may rely on inter-channel error dependencies. Note that, in contrast to vision anomaly detection where an input has a fixed number of three RGB channels, multivariate time-series data can have hundreds of interacting sensor measurements over time. In this work, we show that the choice of scoring function can greatly impact reported model performance, sometimes even more than the choice of TAD DL model itself. To illustrate this, we show in Fig. 2 (A) how different scoring functions (on the x-axis) impact area-under-precision-recall (AUPR) metrics for several DL models on the SWAN [11] dataset. More extensive results are presented and discussed in Section 4.1. In later sections, we propose a simple novel scoring function based on a lightweight ML algorithm (Isolation forest) to map errors to univariate scores, leading to considerable gain in average performance.

Thresholding: Converting anomaly scores into binary anomaly labels is done by choosing a thresholding function that will process test (or validation data) to find a suitable single threshold or thresholds (in case of dynamic approaches). Such optimal thresholding is faced with similar challenges: lack of labeled anomaly data, possible distribution shift at test time, etc. The most widely-used thresholding functions for benchmarking [6, 9, 43] therefore rely on ground truth test anomalies. For instance, “BestF” thresholding selects the optimal single-valued threshold that yields the best F_1 score [9] by looping through sometimes thousands of values and evaluating repeatedly over all test-labels. Similarly, “TopK” selects the threshold resulting in exactly K time-points being labeled as anomalous [43] based on knowledge of how many anomalies should be present at test-time. These thresholding methods tend to yield superior results since they leverage anomaly information in the test data which, for the vast majority of real-world usages, is simply not available to the user. On the other hand, unsupervised thresholding functions can leave a lot to be desired [8, 25] and may be a very interesting future direction for TAD research, particularly when taking into account that many thresholds may need to be defined over time given the periodic phase-aware nature of time-series.

We show that the selection of threshold function, and how much test data is accessed by them, can significantly impact the reported anomaly detection performance. As with the choice of scoring function explained earlier, this variability is often greater than the choice of underlying model as well, as can be seen in Fig. 2 (B): Plot (B.1) displays the performance of an Autoformer model [30] on several TAD datasets when using Top-K thresholding [43], whereas (B.2.) uses SPOT thresholding [25], which is unsupervised. All post-thresholding metrics (F1, F-PA, etc.) radically change between these two. We will present more results in Section 4.2. In the TAD literature, usage and reporting of thresholding functions is inconsistent, making it hard to compare between TAD methods. We attempt to alleviate this shortcoming.

Evaluation Metrics: Anomaly detection models can be evaluated both prior to thresholding (“pre-thresholding”) or after (“post-thresholding”). The former is usually measured via standard modality-agnostic metrics such as AUROC and AUPR. However, their use in TAD literature has been inconsistent, many works only report post-thresholding metrics. We identify this as a major issue since post-thresholding may

be high by virtue of very fine-tuned and supervised threshold selection instead of the underlying TAD model’s contribution. Widespread reporting of pre-thresholding metrics could alleviate this issue. In parallel, the most widespread post-thresholding metrics are: point-wise F1-score and F-PA [32]. The latter is a modified version of the F1-score and is computed after modifying all predictions within an analysis window if at least one point is above the anomaly decision threshold. This requires full access to test-labels since the exact anomaly locations are copied over to the prediction sequence if the previous condition is met. Several works have highlighted serious concerns with the use of F-PA [6, 9, 19]. For instance, [9] shows that when using F-PA, even random score can do better than a fully trained DL model. New post-thresholding metrics have been proposed to alleviate some of the issues with F-PA. For instance, [9] introduce F-PA%K which modifies F-PA to perform point-adjustment only if at least K time-points within a window (number of contiguous anomaly points) are detected as anomaly. [19] introduce “F-K-AUC”, which is the area under the curve of F-PA%K scores when varying K from 0 to the length of the evaluation window. [6] propose F_{comp} , which is a modified F_1 that uses conventional point-wise precision but replaces pointwise recall with event (window-wide) recall.

2.3 TAD comparative analyses and how we address some of their limitations

Numerous TAD evaluation studies have highlighted serious inconsistencies in the TAD field. As explained in Section 2.2, several works have shown worrying behavior and properties of one of the most common TAD metrics “F-PA” [6, 9]. Others have attempted to propose novel metrics that alleviate some of the issues observed in F-PA [6, 9, 19]. Finally, many comparative works show that DL TAD models fail to display consistent performance gain over simple ML or statistical algorithms [21, 22, 28]. Yet, despite the significant contributions of the previous works, a central analysis that has been overlooked so far is how the choice of scoring and thresholding impact model performance. We will show in Section 4 that performance is very sensitive to both of those choices, often even more than the choice of underlying TAD model itself. Moreover, even though previous works have underscored issues with widespread metrics such F-PA, they have not comprehensively quantified how F-PA fares compared to more novel proposed metrics using a large and diverse range of TAD model types. For instance, F-PAK and F-K-AUC were originally tested using a single DL-based autoencoder [9, 19]. To the best of our knowledge, we are the first to quantify cross-metric correlations (both pre and post-thresholding) through diverse TAD model types, seeking to establish a concrete guideline about metric robustness: which metrics should be consistently used to benchmark and which are too problematic to use. This involves correlating performance trends between pre-thresholding and post-thresholding metrics as well.

3 TAD Evaluation Framework (TADEF)

Fig. 1 presents our comprehensive TAD evaluation framework (TADEF) which consists of four main modules: (i) anomaly prediction model, (ii) scoring model, (iii) thresholding function, and (iv) evaluation. Connecting these four modules is key in achieving

robust evaluation of TAD algorithms. TAD prediction model f takes time-series data $X \in \mathbb{R}^{d \times t}$ to calculate the model output $f(X) \in \mathbb{R}^{d \times t}$. The next module, scoring model g , transforms model output into anomaly scores $g(f(X)) \in \mathbb{R}^t$. After this step, we perform pre-thresholding evaluation based on AUROC and AUPR. Given anomaly scores, the thresholding function T then converts anomaly scores into binary labels $T(g(f(X))) \in \mathbb{R}^t$ which indicates if a time point t is anomaly (1) or not (0). Ultimately, post-thresholding evaluation is performed by comparing true anomalies with the predictions based on various metrics, e.g., F_1 , F-PA, F-PAK, F-K-AUC, F-comp.

3.1 Anomaly Prediction Models

We include and analyze 21 algorithms consisting of a mix of DL and traditional ML methods. We include DL solutions from 3 main learning paradigms, i.e. reconstruction-based, contrastive-based and adversarial-based.

1. **Reconstruction-based DL:** One of the main class of algorithms for TAD, these models have an encoder-decoder architecture where the encoder maps input data (non-anomalous) to a latent representation, and the decoder re-maps this latent to reconstruct the input. Learning occurs by minimizing a reconstruction error and the latter is used as an anomaly score during inference. In this category, we include: GPT2 [42], iTransformer [14], DLinear [38], PatchTST [17], MICN [29], TimesNet [31], Crossformer [40], LightTS [39], Informer [41], AutoFormer [30], Reformer [10], Transformer [27], and LSTM-AE [16].
2. **Contrastive-based DL:** These methods learn data representations by contrasting between positive (either same sample or class with some augmentation) and negative samples (different sample or class). Positive samples are trained to have similar representations, and negative samples are pushed apart. We compare two recent contrastive models, TS2Vec [37] and DCdetector [35].
3. **Adversarial-based DL:** Algorithms that leverage adversarial-based training as either their primary or auxiliary training loss/model. From this learning category, we use USAD [3] and AnomalyTransformer [33].
4. **Traditional ML:** These are traditional (non-deep-learning) models. We include four widely used TAD ML methods: Principal Component Analysis (PCA) [15], Isolation Forest (IF) [13], One-class Support Vector Machine (OCSVM) [23], and Local Outlier Factor (LOF) [4].

3.2 Scoring Functions

To address scoring function sensitivity and sub-optimality outlined in Section 2.2, we introduce a new scoring function that uses a traditional and lightweight ML model to map error values to univariate scores. Besides, we slightly modify MeanCh to calculate maximum value over channels. We also include a Gaussian-based scoring function [16].

1. **ML-based scoring:** We use Isolation Forest (IF) [13] to perform the mapping from DL output to univariate anomaly scores. Although in principle, any other lightweight SSL ML method could also work. We train IF with the DL model’s output (validation data). This approach strives to preserve implicit anomaly information across output channels.

2. **Mean over Channels (MeanCh)** [31]: MeanCh simply takes DL model output $f(X)$ and then calculates the mean over output channels. This is the most widely used scoring function in TAD literature.
3. **Max over channels (MaxCh)**: Similar to MeanCh, MaxCh calculates the maximum error over all channels instead of averaging them.
4. **Gaussian scoring** [16]: Gauss-S assumes a Gaussian distribution for the DL model output. We train it on validation data. Anomaly scores are computed as follows:

$$g(f(X)) = \sum_{i=1}^d -\log(1 - \Phi(\frac{f(X) - \hat{\mu}}{\hat{\sigma}})) \quad (1)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the empirical mean and standard deviation and Φ is the cumulative distribution function (cdf) of $\mathcal{G}(0, 1)$.

3.3 Thresholding

For thresholding, we create two categories based on the availability of test anomalies:

1. Access to Test Labels/Meta: In this category, optimal threshold is discovered by leveraging labeled test anomalies. Thresholding methods in this category are the most prevalent in the TAD literature. We include BestF [26] and Top-K [43]:

- BestF discovers a threshold value that maximizes the F-score. The set of threshold values is created based on *precision recall curve*. To illustrate the risk of relying on test label access, we propose two variations of BestF: (i) BestF partial access (PA) where we limit 10% anomaly access (both anomalous and normal data-points), and (ii) BestF full access (FA), which utilizes all test-data.
- Top-K sets the threshold value to select the highest-scoring $K\%$ of points as anomalous, according to knowledge (meta-information) of the ground truth anomaly ratio (K) in the test set.

2. Unsupervised: A more realistic approach, unsupervised thresholding does not rely on ground-truth anomalies. Here, we consider the following methods:

- Validation Percentile (ValPer) simply uses validation data to find an optimal threshold. Since validation data for SSL TAD includes only normal data, high test scores w.r.t validation scores can be suspected of anomaly. We set the threshold to a top percentile of the validation data, i.e. a percentile corresponding to two or three standard deviations from the mean.
- SPOT and DSPOT [25] are based on the Peak Over Threshold (POT) model, which uses the Pickands-Balkema-de Haan theorem to model extreme values. While Streaming Peak over Threshold (SPOT) is proposed for streaming data, Streaming Peak over Threshold with drift (DSPOT) updates the mean value every several steps to account for a possible drift in the dataset.
- Dynamic Thresholding (DynTh) [8] is a non-parametric thresholding technique that calculates reconstruction error for each time step and applies exponential weighted moving average (EWMA) to generate smoothed errors. The threshold value is set to that which causes the greatest percent decrease in the mean and standard deviation of the smoothed errors.

3.4 Evaluation Metrics

We evaluate anomaly detection performance at two stages of the TAD inference pipeline: Prior to thresholding (pre-thresholding) and after (post-thresholding).

1. **Pre-Thresholding (*PreT*):** We use conventional area under receiver-operating-curve (AUROC) and area under precision-recall curve (AUPR). These metrics are the go-to for anomaly detection in modalities such as vision, but remain inconsistently reported in TAD. By definition, both metrics are agnostic to thresholding function.
2. **Post-Thresholding (*PosT*):** Anomaly detection in practice needs a threshold to obtain anomaly predictions [6]. In this group, the model performance is evaluated after thresholding, and thus, are dependent on the choice of thresholding function. For *PosT*, we create two types of evaluation:
 - Access to Test Labels: The metrics in this group leverage true anomalies to modify anomaly predictions. We consider F_{PA} [32], F_{PAK} [9], and F_{K-AUC} [19]. F_{PA} applies point adjustment (PA) and calculates F_1 score. F_{PAK} uses PA%K protocol with $K = 20$ (default K value) and measures F_1 score. Ultimately, F_{K-AUC} is an improved version of F_{PAK} which calculates F_1 score at different levels of $K \in [0, 100]$, and computes the area under this curve. More details of these metrics can be found in Section 2.2.
 - Unsupervised: We include the conventional point-wise F_1 score [32] and a more recently proposed F_{comp} [6]. The latter uses point-wise precision scores but replaces point-wise recall with event-wide recall (with a fixed-size window of points being considered).

Table 1. TAD Datasets Summary

Dataset	Training	Test	N. channels	Anomalies(%)
SWaT [7]	495K	449K	51	12.14
WADI [2]	784K	172K	123	5.78
SMD [26]	708K	708K	38	4.16
PSM [1]	132K	87K	25	27.74
SMAP [8]	135K	427K	25	12.79
MSL [8]	58K	73K	55	10.55
SWAN [11]	60K	60K	38	32.6
GECCO [11]	69K	69K	9	1.05

3.5 Datasets

Table 1 summarizes the selected TAD datasets in terms of training and test data size, number of channels, and the anomaly percentage in the test portion.

1. **Secure Water Treatment (SWaT):** SWaT represents a realistic cyber-physical system of industrial water treatment plant. The data was collected for 11 days where 41 anomalies were injected in the last 4 days. The dataset includes physical properties and network traffic collected from 51 sensors, e.g., flow meters, level transmitters, conductivity analyzer, and actuators, e.g., motorized valves and pumps.
2. **Water Distribution (WADI):** WADI depicts a water distribution network which includes three control processes, where each controlled by its own set of Programmable Logic Controllers (PLCs). WADI is physically connected to SWaT which supplies filtered water. The dataset consists of data from 123 sensors and actuators collected over 14 days for normal operation and 2 days with 15 attacks.
3. **Server Machine Dataset (SMD):** SMD includes server machine metrics, with 38 features such as memory usage and CPU utilization. The data measures 5-weeks in length where half is used for training, and the other half is labeled for testing.
4. **Pooled Server Metrics (PSM):** PSM is collected from application server nodes at eBay. The dataset consists of 26 features to represent server machine metrics. The training set includes 13 weeks, followed by eight weeks for testing. Although anomalies are present in both training and test set, only the latter is labeled.
5. **Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL):** SMAP and MSL are real-world datasets collected from a NASA spacecraft. These data are from an incident surprise anomaly report for a spacecraft monitoring system. While SMAP includes 25 features, MSL represents 55 various channels.
6. **NIPS-TS-SWAN (SWAN):** SWAN is a comprehensive, multi-variate time series benchmark extracted from solar photospheric vector magnetograms in Spaceweather HMI Active Region Patch series. SWAN has more than 32% anomalies, making it the least realistic TAD dataset among the selected ones.
7. **NIPS-TS-GECCO (GECCO):** GECCO is a drinking water quality dataset for the Internet of Things (IoT). GECCO only includes nine features, making it the anomaly dataset with the lowest dimension.

3.6 TAD Baselines

[9] suggested establishing a strong new baseline for TAD evaluation based on a random-weighted DL model. [21] later added a few more trivial but effective baselines. Similar to random guess for a classification task, newly proposed TAD methods should outperform these baselines to demonstrate model effectiveness. We analyze 4 TAD baselines:

1. **Random anomaly scores (B_{rand}):** B_{rand} uses anomaly scores drawn from a uniform distribution $\sim \mathcal{U}(0, \max(g(f(X_{test})))$ where $\max(g(f(X_{test})))$ denotes the maximum anomaly score in the test data.
2. **Raw input as anomaly scores (B_{input}):** B_{input} is the baseline where the input data is copied as anomaly scores.
3. **L2-norm scores (B_{norm}):** B_{norm} calculates the L2-norm of the raw input.
4. **Untrained model anomaly scores (B_{untr}):** B_{untr} represents an untrained DL model initialized from a Gaussian distribution $\sim \mathcal{N}(0, 0.02)$.

Table 2. Scoring Function Analysis. We compare how different scoring functions alter pre-thresholding performance, as measure via AUROC and AUPR. Our ML-based scoring function (first column) shows significant improvement over the usual scoring method MeanCh. Last column **Std-Dev** computes the standard deviation that can result from changing scoring function.

Dataset	ML (IF)		GS		MeanCh		MaxCh		Std-Dev	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
WADI	71.32	15.42	65.21	34.07	54.04	6.82	52.51	6.58	7.82	11.18
SMD	79.09	21.74	72.71	14.9	77.78	18.29	76.02	16.9	2.39	2.49
MSL	64.67	16.35	57.72	15.47	66.91	19.54	66.92	19.96	3.77	1.95
SWAN	87.12	80.12	85.06	76.92	86.37	76.88	84.15	74.17	1.15	2.11
PSM	78.84	54.24	71.31	51.89	73.54	52.72	70.88	52.92	3.17	0.84
SMAP	50.58	13.47	56.08	13.84	58.79	15.47	58.15	15.76	3.23	0.99
SWAT	83.57	71.82	77.12	29.11	83.23	72.9	81.4	71.26	2.57	18.58
GECCO	94.97	32.42	81.45	38.46	93.56	32.77	93.27	32.82	5.44	2.51
AVG	76.27	38.19	70.83	34.33	74.28	36.92	72.91	36.29	1.98	1.40

4 Results

4.1 Pre-Thresholding TAD Performance

Comparison across SOTA: We first analyze pre-thresholding TAD performance of several DL, ML and baseline methods, as presented in Fig. 3. We employ both AUROC and AUPR which are the standard pre-thresholding metrics for evaluating anomaly detection in data modalities such as Vision. The advantage of comparing performance at a pre-thresholding stage is the disentanglement of the TAD model’s performance from its choice of thresholding function. The reason being that thresholding is in itself a very challenging problem, as discussed in Section 2.2. In fact, as we will show in Section 4.2, the choice of thresholding function starkly alters the performance of all tested post-thresholding metrics. Despite clear benefits, use of pre-thresholding metrics has been inconsistent in TAD literature. In Fig. 3, row-wide panels display AUROC (top) and AUPR (bottom) scores for all eight TAD datasets and for each TAD model. The inter-quartile distribution per model is shown as box-plots, with the “average” performance (across datasets) designated as a horizontal dash. Individual dataset performances are displayed by scatter plots with a unique symbol, e.g., circle, square, per dataset as indicated in the legend. The color coding used designates, from left to right: DL models (beige), traditional ML models (salmon) and baseline models (purple).

Overall, clear trends emerge from the results of Fig. 3: (1) DL models (beige) do not consistently outperform traditional ML (salmon) or even simple baselines (purple). This is observed with both AUROC and AUPR; (2) The performance variability of most models across different datasets is significant. This is especially the case for AUPR scores and most likely due to its sensitivity to data imbalance, i.e., low anomaly

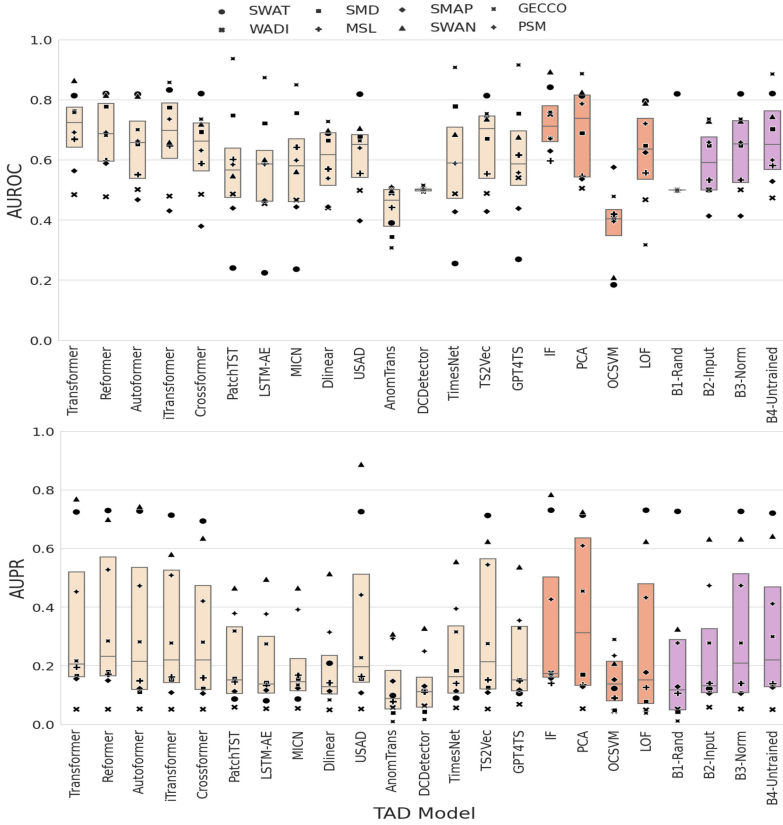


Fig. 3. SOTA Comparison across TAD models with Pre-thresholding metrics

ratios. For instance, datasets with high anomaly ratios such as SWAN and PSM tend to have higher AUPR values than datasets with low ratios such as WADI and GECCO, vid. Table 1; (3) Performance trends between AUROC and AUPR vary significantly. Models that score best with AUROC are not always winners with AUPR, and vice versa. Most likely this occurs because AUROC, as a metric, mitigates the impact of data imbalance. Yet, since imbalance is inherent to anomaly detection applications in the real-world, we argue that AUPR should be prioritized over AUROC for more accurate TAD evaluation. To see this relationship quantitatively, we will analyze inter-metric performance correlation in Section 4.2. For the remainder of the paper, we select the best performing model (as measured by combined AUPR and AUROC) per dataset: MICN (SMD), Autoformer (WADI and SWAN), Transformer (MSL and SMAP), LSTM-AE (PSM), Reformer (SWAT) and PatchTST (GECCO).

Choice of Scoring function: Table 2 compares different scoring functions. We report results for the best performing DL model per dataset as defined previously. We can

observe that the choice of the scoring often alters performance (vid. column [Std-Dev](#)) and that overall, there is a gain from applying our non-trivial (ML) scoring. On WADI, our ML-based scoring improves over MeanCh AUPR by 127%. DL Models which directly map to univariate scores or that deploy more refined scoring are an interesting direction for future TAD research. For the following section, Post-Thresholding Analysis, we use the best model per dataset, combined with the best-performing scoring function per dataset: ML-scoring (WADI, SMD, SWAN, PSM, GECCO), MeanCh (SWAT), MaxCh (MSL, SMAP).

4.2 Post-Thresholding TAD performance

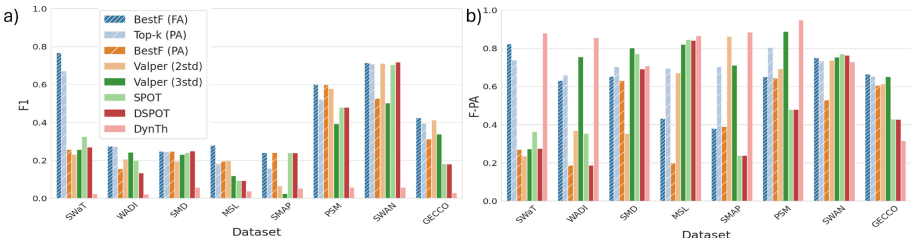


Fig. 4. Post-thresholding performance is highly susceptible to the thresholding function used. Notably, F-PA (right-b) overestimates performance compared to F1 (left-a). For instance, thresholding functions with very poor F1 scores such as DynTh, climb to first place with F-PA.

Choice of Thresholding function: Fig. 4 shows how post-threshold metrics (F1, F-PA by row) are highly sensitive to the choice of thresholding function (shown as a different colored bar per threshold type). The texture on each bar designates if that particular thresholding function requires access to test-labels (Full Access “FA” with less spaced dash), partial access (PA, with more spaced dash) and finally, no access to test-labels (solid colored bars)¹. Based on the F1 metric, “BestF” is consistently the best thresholding function, requiring access to test labels, and is the most widely used across TAD literature. It significantly outperforms by looping through all possible thresholds and selecting the one with the highest performance, evaluated using the entire labeled test set. While from an evaluation perspective this may be acceptable, it is far from realistic, since most TAD applications are fully unsupervised. To further exemplify the danger of relying on test-label-access, we introduce a modified BestF function which only uses 10% of the test-data (both anomalous and normal data-points). In this case, performance drastically declines w.r.t BestF full-access. In sum, caution is needed when comparing post-thresholding performance across TAD literature because authors do not always clearly disclose what thresholding mechanism is used. It is unfair to compare,

¹ High-resolution figures included in supplementary for better clarity

for example, Full-Access and No-Access scores. In Supplementary, we include more extensive test-access variation plots, varying PA ratio from 0-100.

Lastly, in Fig. 4 we also show how F-PA overestimates performance across all datasets and most thresholds (comparing panel a and b). For instance, take the worst performing threshold according to the F1 metric, e.g. DynTh. Both its recall and precision are very poor. Yet, with F-PA it unjustifiably climbs to the best performing thresholding function. F-PA has been shown [9,28] to yield unreliable scores and here we again emphasize this. In the next section, we quantitatively measure how each metric relates to one another and which should be dropped from future TAD literature.

Table 3. Pearson correlation coefficient (PCC) among different TAD evaluation metrics, both pre-thresholding (AUPR, AUROC) and post-thresholding (F1, F-comp, F-K-AUC, F-PAK, F-PA).

Pearson-C	F1	F-comp	F-K-AUC	F-PAK	F-PA	AUROC	AUPR
F1	1	0.633	0.413	0.424	0.115	0.466	0.726
F-comp	0.633	1	0.103	0.12	0.351	0.294	0.22
F-K-AUC	0.413	0.103	1	0.989	0.551	0.423	0.682
F-PAK	0.425	0.12	0.989	1	0.555	0.441	0.645
F-PA	0.115	0.351	0.551	0.555	1	0.054	0.179
AUROC	0.466	0.294	0.423	0.441	0.054	1	0.576
AUPR	0.726	0.22	0.682	0.645	0.179	0.576	1

Table 4. Correlation values. Effect of different thresholding styles on the Pearson correlation between pre-thresholding metrics (AUROC, AUPR) and post-thresholding metrics (F1, F-K-AUC).

Post-Th	Pre-Th	BestF	Top-k	BestF-PA	Valper	Valper	SPOT	DSPOT	DynTh
				(10%)	(2 σ)	(3 σ)			
F1	AUROC	0.604	0.684	0.432	0.668	0.815	0.361	0.348	-0.182
	AUPR	0.981	0.979	0.679	0.718	0.733	0.823	0.784	0.114
F-K-AUC	AUROC	0.629	0.695	0.438	0.661	0.766	0.445	0.416	-0.662
	AUPR	0.976	0.976	0.559	0.654	0.654	0.819	0.751	0.067

What metrics should we keep? In this section, we compare all pre-thresholding and post-thresholding metrics with the goal of identifying metrics that are consistent indicators of performance (pre and post-thresholding). The lack of consistency among metrics such as F-PA is concerning and here we aim to rigorously quantify the extent of this inconsistency. Table 3 contains the correlations between all evaluation metrics tested in this paper. The color code indicates that greener cells have higher positive Pearson

correlation, whereas redder cells have lower correlation. For post-thresholding metrics, correlation values between each metric pairs (cell) is computed by averaging the individual correlations between all thresholding functions for all models and datasets. From these results, it is evident that F-PA has the worst collective correlation with all other metrics, including both pre-thresholding metrics (AUROC and AUPR). Alternatively, pointwise F1 is the post-thresholding metric with highest correlation to the pre-thresholding metrics, which emphasizes its enduring reliability, even when compared to newer metrics such as F-PAK and F-K-AUC. Finally, these results also highlight that AUPR consistently shows higher correlations with post-thresholding scores than AUROC, suggesting it should be preferred as the pre-thresholding metric in TAD.

In Table 4, we show results for F1 and F-K-AUC across all tested thresholding functions. Results are averaged across all eight datasets. The main takeaways are: (1) As expected, full and meta-access thresholds (BestF, Top-K) yield consistently higher correlations to pre-th metrics, especially AUPR. (2) In contrast, correlations clearly indicate that thresholds such as DynTh or BestF (PA) perform poorly. (3) In summary, Table 4 underscores that thresholding is a difficult problem in itself and future work should devote more attention to developing better unsupervised thresholding techniques.

5 Conclusion

Through our comprehensive analysis of SOTA TAD models and evaluation metrics, several important conclusions can be drawn: (1) Firstly, there is a significant need for more robust TAD models. The latest algorithms, which rely on complex DL techniques, fail to show consistent improvement over trivial baselines and simple ML methods; (2) Secondly, the central reason for this stagnated TAD progress is reliance on faulty metrics, e.g. F-PA, lack of pre-thresholding score reporting and, inconsistent use and reporting of scoring or thresholding functions. All of those combined, contribute enormously to model performance scores and can be handpicked to boost performance at the expense of reliable comparative benchmarking; (3) Additionally, the periodic, phase-dependent and high-dimensional nature of multivariate time-series is far from trivial. Current unsupervised thresholding techniques are not on-par to handle this, which is an opportunity for future work. (4) We propose more reliable variants of currently used scores (e.g. ML-scoring) and metrics (e.g. BestF-PA) as well as comprehensively analyze existing metric correlations where we show that AUPR (pre-thresholding) and point-wise F1 (post-thresholding) metrics are still the most consistent and robust. Future work can leverage this comparative framework to propose additional robust evaluation metrics.

Acknowledgements. This work has been funded in part by NSF, with award numbers #1826967, #1911095, #2003279, #2052809, #2100237, #2112167, #2112665, and in part by PRISM and CoCoSys, centers in JUMP 2.0, an SRC program sponsored by DARPA.

References



1. Abdulaal, A., et al.: Practical approach to asynchronous multivariate time series anomaly detection and localization. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining (2021)

2. Ahmed, C.M., et al.: Wadi: a water distribution testbed for research in the design of secure cyber physical systems. In: Proceedings of the 3rd international workshop on cyber-physical systems for smart water networks (2017)
3. Audibert, J., et al.: Usad: Unsupervised anomaly detection on multivariate time series. In: 26th ACM SIGKDD international conference on knowledge discovery & data mining (2020)
4. Breunig, M.M., et al.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data (2000)
5. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning (2006)
6. Garg, A., et al.: An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems* (2021)
7. Goh, J., et al.: A dataset to support research in the design of secure water treatment systems. In: Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11. Springer (2017)
8. Hundman, K., et al.: Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (2018)
9. Kim, S., et al.: Towards a rigorous evaluation of time-series anomaly detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
10. Kitaev, N., et al.: Reformer: The efficient transformer. [arXiv:2001.04451](https://arxiv.org/abs/2001.04451) (2020)
11. Lai, K.H., et al.: Revisiting time series outlier detection: Definitions and benchmarks. In: Conference on neural information processing systems datasets and benchmarks track (2021)
12. Lindstrom, M.R., et al.: Functional kernel density estimation: Point and fourier approaches to time series anomaly detection. *Entropy* (2020)
13. Liu, F.T., et al.: Isolation forest. In: IEEE international conference on data mining (2008)
14. Liu, Y., et al.: itransformer: Inverted transformers are effective for time series forecasting. [arXiv:2310.06625](https://arxiv.org/abs/2310.06625) (2023)
15. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (pca). *Computers & Geosciences* (1993)
16. Malhotra, P., et al.: Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016)
17. Nie, Y., et al.: A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022)
18. Park, D., et al.: A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* (2018)
19. Pintilie, I., et al.: Time series anomaly detection using diffusion-based models. In: 2023 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE (2023)
20. Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis (2014)
21. Sarfraz, M.S., et al.: Position paper: Quo vadis, unsupervised time series anomaly detection? *arXiv preprint arXiv:2405.02678* (2024)
22. Schmidl, S., et al.: Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment* (2022)
23. Schölkopf, B., et al.: Support vector method for novelty detection. *Advances in neural information processing systems* (1999)
24. Shyu, M.L., et al.: Principal component-based anomaly detection scheme. *Foundations and novel approaches in data mining* (2006)
25. Siffer, A., et al.: Anomaly detection in streams with extreme value theory. In: 23rd ACM SIGKDD international conference on knowledge discovery and data mining (2017)

26. Su, Y., et al.: 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 (2019)
27. Vaswani, A., et al.: Attention is all you need. *Advances in neural information processing systems* (2017)
28. Wagner, D., et al.: Timesead: Benchmarking deep multivariate time-series anomaly detection. *Transactions on Machine Learning Research* (2023)
29. Wang, H., et al.: Micn: Multi-scale local and global context modeling for long-term series forecasting. In: The Eleventh International Conference on Learning Representations (2022)
30. Wu, H., et al.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* (2021)
31. Wu, H., et al.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: The eleventh international conference on learning representations (2022)
32. 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 (2018)
33. Xu, J., et al.: Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint [arXiv:2110.02642](https://arxiv.org/abs/2110.02642)* (2021)
34. Yang, Y., et al.: Pipeline safety early warning by multifeature-fusion cnn and lightgbm analysis of signals from distributed optical fiber sensors. *IEEE Transactions on Instrumentation and Measurement* (2021)
35. Yang, Y., et al.: Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2023)
36. Yeh, C.C.M., et al.: Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th international conference on data mining (ICDM). *Ieee* (2016)
37. Yue, Z., et al.: Ts2vec: Towards universal representation of time series. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
38. Zeng, A., et al.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence (2023)
39. Zhang, T., et al.: Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint [arXiv:2207.01186](https://arxiv.org/abs/2207.01186)* (2022)
40. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting (2022)
41. Zhou, H., et al.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence (2021)
42. Zhou, T., et al.: One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* (2023)
43. Zong, B., et al.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)



The Analog Layer: Simulating Imperfect Computations in Neural Networks to Improve Robustness and Generalization Ability

Giovanni Maria Manduca, Antonino Furnari^(✉) ,
and Giovanni Maria Farinella 

Department of Mathematics and Computer Science, University of Catania, Catania,
Italy

MNDGNN02C20I754X@studium.unict.it,
{antonino.furnari,giovanni.farinella}@unict.it
<https://iplab.dmi.unict.it/fpv/>

Abstract. Usage of noise is common at the input level of neural networks as a means of data augmentation. This study examines the impact of incorporating stochastic noise deeply into the activation signals between layers of neural networks, simulating analog circuit computation. We introduce the “Analog Layer” model, which embeds inherent stochasticity in the computation of activations and develop an algorithm to dynamically adjust noise levels during training, thus creating a noisy yet controlled curriculum learning training environment. We evaluate our approach on Fully Connected and Convolutional Networks using the MNIST, FashionMNIST, CIFAR10, and CIFAR100 datasets. The proposed framework is assessed considering accuracy, robustness to input and state perturbations, resistance to FSGM adversarial attacks and feature map entropy. We show that our method can improve the network’s base accuracy, as well as its resilience to input and state perturbations and adversarial attacks. The proposed approach allows to compute representations which have a lower distribution entropy across its neurons, allowing to achieve improved robustness. We finally give an interpretation of the proposed technique as both a regularization method and a consensus mechanism.

Keywords: Neural Networks · Deep Learning · Neural Network robustness · Stochastic Noise · Curriculum Learning

1 Introduction

The usage of noise during training in neural network architectures is well-studied in the context of deep learning. The application of stochasticity has generally

G. M. Manduca—This research has been supported by the project Future Artificial Intelligence Research (FAIR) – PNRR MUR Cod. PE0000013 - CUP: E63C22001940006.

proven itself to be effective in improving the convergence speed, generalization ability and overall quality of the model [1]. Usually, noise is introduced during training by applying perturbations in the form of data augmentation [7], either by employing known, domain-specific transformations or by introducing noise directly into the input either by sum or multiplication. Perturbations can also be implemented within the training algorithm itself, as for example with learning rate schedulers [2].

It is well-known that the behavior of neural networks can be significantly influenced by minor modifications to their inputs [3]. These small perturbations accumulate error layer by layer, leading to an avalanche effect which substantially changes at deeper levels of the network’s representation, ultimately leading to drastic changes to the output. The processes of crafting these malicious inputs and their respective countermeasures have emerged as a major area of interest in machine learning and particularly in computer vision, where humans often cannot perceive the differences between the original and adversarial images [5].

We propose a novel approach characterized by the introduction of stochasticity in the computations of the internal representations of the network. We study how noise in the internal mathematical operations of the neural network affects the training of neural networks, postulating that this challenging source of noise can improve model robustness and generalization ability. By injecting noise directly into the computation of the activations, our approach introduces perturbations at the internal representation level, thus acclimating the network to the presence of noise, thereby making adversarial attacks more challenging.

Furthermore, while many successful methods such as Dropout [6] considered a constant level of noise, our method uses a non-constant level of noise throughout the training, which is regulated to increase the difficulty of the task accordingly with the status of the training, following a curriculum learning recipe.

The contributions of this paper are threefold:

- We propose a novel approach to inject noise during the training of neural networks. Our method embeds noise deeply into the computation of the internal activations of the model, which results in a more challenging setting for learning robust representations.
- We thoroughly evaluate our approach on classification tasks on the MNIST, FashionMNIST, CIFAR10 and CIFAR100 datasets using MLPs and ConvNets. Results show that our method can not only increase baseline performance in classification, but also lead to an inherently increased robustness to input perturbations, state perturbations and adversarial attacks. Furthermore, we show that the introduced Analog Layers produce a weights distribution that is less entropic than their analog counterparts and better distribute information across their neurons.
- We propose an algorithm to inject noise progressively, following a curriculum learning method, and we give an interpretation of our method as both a consensus and regularization method.

2 Related Work

Our research is related to and inspired by previous works and investigations on data augmentation, hyperparameter scheduling and more generally the usage of noise in the training process.

2.1 Data augmentation as means to improve performance

Data augmentation encompasses a range of techniques used to mainly increase the number and variety of training data, leading to an improvement in both performance and generalization [8]. These approaches, both domain-specific and agnostic, range from simple techniques like shifting, cropping, flipping, and color jittering to more advanced machine learning based methods. The latter often make use of unsupervised algorithms such as VAEs, GANs and diffusion models [9] to train a model capable of generating new samples from the input distribution, which are then used as new data points. These approaches have proven particularly effective in low-data scenarios like medical imaging [11, 12]. The proposed analog layer can be seen as a deeper form of potentially domain-agnostic data augmentation, as well as a deeper representation level augmentation.

2.2 Hyperparameter scheduling to improve convergence rate

Hyperparameter scheduling is a widely utilized technique, with the most notable examples being learning rate scheduling [2, 13] and batch size scheduling [14]. Effective hyperparameter scheduling can significantly boost the convergence rate [13] and overall performance of a model. This improvement can be often related to scheduling’s ability to mitigate, overcome or avoid problematic/pathological phases of training, as well as to introduce a certain level of stochasticity during training. Furthermore, hyperparameter scheduling can serve as a form of curriculum learning, progressively increasing task difficulty throughout the training process. In the context of this research, we propose two algorithms designed to vary a specific hyperparameter (namely, the global noise level) that directly correlates with training difficulty.

2.3 Insertion of noise to improve robustness

The incorporation of noise and stochasticity is ubiquitous in machine learning and deep learning research. Introducing noise not only in input data but in intermediate layers and output labels as well is a strategy used to enhance robustness and generalization. A famous example of this is Dropout [6], together with its many modifications such as Adaptive Dropout [15], different sampling strategies [16] or generalizations [17]. Stochastic processes are also employed in Variational Autoencoders [18], which achieve disentanglement of the latent space through sampling. Moreover, techniques such as label smoothing [19], which implicitly inject noise in the training process by providing more uncertain labels, have

been explored and found to be effective in manipulating training labels. The integration of noise as a fundamental component of the training procedure has been shown to be an effective means of improving neural network performance, as demonstrated by Denoising Autoencoders [20] and further confirmed by Diffusion Models [21], which are widely recognized as state of the art for many generative tasks. Other stochastic techniques, such as Stochastic Depth [22] and stochastic ensemble learning [23], can be regarded as noise injection techniques in the learning process. Numerous studies have investigated the development of training algorithms tailored to specific noise levels [24]. As for our research, we focus instead on designing a training regimen that adaptively adjusts the noise level throughout the training process in order to maximize performance and robustness.

2.4 Adversarial attacks and defence techniques

Neural networks are notoriously susceptible to input state perturbations. This sensitivity constitutes a weak point for adversaries to exploit. This weakness was first highlighted in [3] and has since evolved into a very active research field, with many types of attacks and defenses [4].

In this work, we will make use of a simple type of adversarial attack, called Fast Sign Gradient Method (FSGM), proposed in [5]. FSGM is a black-box attack that uses the gradient of the loss function in order to generate an adversarial perturbation.

3 Methodology

In this section we introduce the Analog layer framework, as well as the scheduling algorithms for the global noise variation.

3.1 Analog Layer

Taking inspiration from analog circuits, which add inherent stochasticity in every computation due to their analog nature, we propose an ‘‘Analog Model’’ as a modification to a standard neural network layer directly injecting noise into the computation of its activations. We consider a parametric model h_θ , a loss function L , a training algorithm, a global noise intensity α and a noise variation (scheduling) algorithm. In particular, h_θ is a model composed of at least one analog layer. To avoid direct noise influence on the output labels, we don’t use an analog layer as the output layer. Given a generic parametric function f_θ we define its analog counterpart $f_{\alpha,\beta,\theta}^{\text{analog}}$ as:

$$f_{\alpha,\beta,\theta}^{\text{analog}} = N(f_\theta(x), (\alpha\beta)^2) \quad (1)$$

where β and α are hyperparameters called layer-wise noise intensity and global noise intensity. In practice, we prefer to consider the following equivalent definition:

$$f_{\alpha,\beta,\theta}^{\text{analog}} = f_\theta(x) + \alpha\beta\lambda \quad (2)$$

where $\lambda \sim N(0, 1)$ is an unit Gaussian tensor of appropriate shape. This reparametrized version exposes the deterministic function for backpropagation. Note that λ is sampled each time we calculate the function, and both α and β are not learnable.

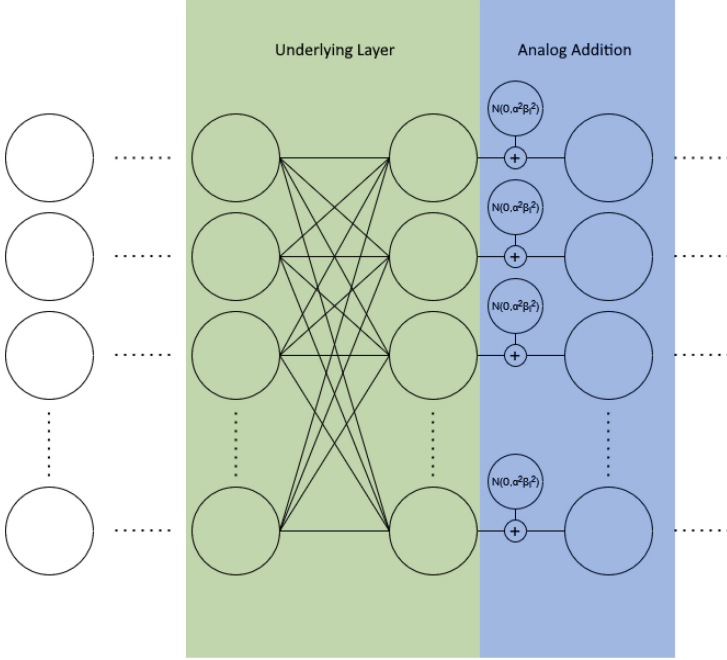


Fig. 1. A visualization of an analog layer. The variance of the Gaussian depends on both α and β_i .

Note that, in the context of an analog model, each analog layer has its own value of β , which we denote as β_1, \dots, β_n , while α is shared among all layers. In the following sections, whenever we refer to the value of a generic β_i in the context of a generic Analog Layer, we will refer to it simply as β . Figure 1 gives an illustration of the structure of a generic analog layer.

3.2 The role of α and β

Before jumping to the more practical parts of establishing an analog training model, such as choosing a scheduling algorithm or the hyperparameter values, it is important to understand the role of α and β . The main objective of our research was to develop a method for adaptively introducing noise into the training process, adjusting the level of noise in response to the task’s difficulty at a given point during the training process. Our model was inspired by the computational paradigm of analog chips, which trade off precision for enhanced speed and

efficiency due to their analog nature. Specifically, we drew an analogy between these noisy computations and the introduction of noise into our training process. Each operation executed by an analog chip was conceptualized as introducing a certain degree of noise to the overall computation. The role of α and β is to capture this intuition: α represents the quality of our hypothetical circuit, that is, the amount of noise injected by each operation, while β models the number of computations performed by each layer. This rationale also underpins our decision to assign distinct values β_1, \dots, β_n in each layer, as the computational complexity of each layer is dependent on the underlying function being computed. In contrast, if we assume that the level of noise introduced by the circuit for individual operations remains consistent across all layers of the network, we can simplify the noise variation problem, since we only need to schedule α , our global noise level.

3.3 Fully Connected and Convolutional Analog Layers

The proposed definition of analog function makes it possible to construct the analog variant of any parametric function f_θ . In this paper, we focus on the analog variant of (arguably) the two most popular parametrized layers in neural networks: Fully connected Layers and Convolutional Layers. The Fully Connected Analog layer is defined as follows:

$$AFC_{\alpha,\beta,\mathbf{W},\mathbf{b}} = \mathbf{W}x + \mathbf{b} + \alpha\beta\lambda \quad (3)$$

whereas the Analog Convolutional Layer is defined as:

$$AC_{\alpha,\beta,\mathbf{W},\mathbf{b}}(x) = \mathbf{W} * x + \mathbf{b} + \alpha\beta\lambda \quad (4)$$

3.4 Choosing β for AFC and AC

In light of what we said in the previous paragraph, it is evident that we want the value of β to be some kind of function of the input and output sizes which defines the complexity of the function itself. In particular, since we want to add noise for each computation, we investigate different methods of setting β as a function of the input and output dimensions of the layer. In particular, for an *AFC* layer with input dimension n and the output dimension m and for an *AC* with n input channels, m output channels and kernel of size $a \times b$, we studied the strategies reported in Table 1. After determining β_1, \dots, β_n , we normalized their values as follows:

$$\beta_i^{\text{normalized}} = \frac{\beta_i}{\min_k \beta_k} \quad (5)$$

This makes it possible to train different strategies with relatively similar hyperparameters of the Noise Variation Scheduling Algorithms described in the following section.

Table 1. The strategies we used to choose β and the rationale behind.

Strategy	AFC	AC	Rationale
Constant	$\beta = 1$	$\beta = 1$	Value non dependant on the shapes of input and output
Square Root	$\beta = \sqrt{n}$	$\beta = \sqrt{n \cdot a \cdot b}$	Expectation of the noise level for <i>input_size</i> operations.
Logarithmic	$\beta = \log(n)$	$\beta = \log(n \cdot a \cdot b)$	Logarithmic dependence on the input size
Mass	$\beta = \frac{n}{m}$	$\beta = \frac{n \cdot a \cdot b}{m}$	Reduce noise when input comes from a big (noisy) layer

3.5 Global Noise (α) Variation Scheduling

We now tackle the problem of developing an algorithm that can dynamically adjust the value of α . Ideally, we want our algorithm to make α as high as possible without degrading the network’s performance G (as expressed by some performance measure function g). Moreover, the algorithm should be able to respond to changes in the network performances and adapt the noise level accordingly. The algorithm should be cautious in increasing the level of noise and overreacting when correcting a level that is too high, since we want to avoid our model to go into catastrophic forgetting [25] of the original non-noisy task. Finally, the algorithm should be capable of autonomously determining the maximum noise level, which would eliminate the need for any guesswork or tuning. We also want to avoid making the training too hard at the start by not giving the training algorithm a chance to take a good path. We present and test two scheduling algorithms, namely Additive Increment Additive Decrement and Additive Increment Multiplicative Decrement. In particular, the latter borrows a concept (namely the multiplicative decrease) from congestion avoidance schemes in transmission protocols like TCP [26]. Note that the symbol \gg indicates a total order relationship specific to G .

Additive Increment Additive Decrement (AIAD) The AIAD algorithm is described in Algorithm 1. The algorithm is very simple: we define a zone under which we determine that our algorithm is underperforming ($G < t_{\text{reject}}$), which signals us to reduce the noise level. Similarly, we define a zone above which we determine that our algorithm is overperforming ($G > t_{\text{accept}}$), which signals us that our model is capable of tackling the problem with the current noise level and we can increase it. In the region $t_{\text{reject}} < G < t_{\text{accept}}$, we simply don’t vary the noise, so that our models can train with a constant noise level for however long is necessary to beat t_{accept} (or fail and fall below t_{reject}). Then, we define an increase i and a decrease d . We pick $d \gg i$ to enforce the desired “overreaction” property of the scheduler.

At train time, we first train on w warmup training batches at a fixed noise level α_0 . For the batches after that, we adjust α based on G as determined on the current training batch against t_{accept} and t_{reject} . Finally, we always make sure that α doesn’t become negative by enforcing 0 as α ’s lowerbound.

Additive Increment Multiplicative Decrement (AIMD) The AIMD algorithm (Algorithm 2) builds on the AIAD algorithm by changing only one thing,

Algorithm 1. AIAD algorithm

```

1:  $\alpha_0, w, i, d, t_{\text{accept}}, t_{\text{reject}} \leftarrow$  Scheduling hyperparameters // As described before
2:  $\alpha \leftarrow 0$  // The variable noise level
3: for  $step \leftarrow 1$  to  $w$  do // Warmup phase
4:   [...] // Do training as usual
5: for  $step \leftarrow w + 1$  to  $\infty$  do
6:    $x, y \leftarrow$  next training example
7:    $y' \leftarrow h_{\theta, \alpha + \alpha_0}(x)$ 
8:    $G \leftarrow g(y, y')$  // Get the performance measure
9:   if  $G > t_{\text{accept}}$  then
10:     $\alpha \leftarrow \alpha + i$  // We beat  $t_{\text{accept}}$ 
11:   else
12:     if  $G < t_{\text{reject}}$  then
13:        $\alpha \leftarrow \alpha - d$  // We fall below  $t_{\text{reject}}$ 
14:        $\alpha \leftarrow \max(0, \alpha)$  // Ensure  $\alpha$  is always positive
15:   [...] // Perform the rest of training as usual

```

Algorithm 2. AIMD algorithm

```

1:  $\alpha_0, w, i, d, t_{\text{accept}}, t_{\text{reject}} \leftarrow$  Scheduling hyperparameters // As described before
2:  $\alpha \leftarrow 0$  // The variable noise level
3: for  $step \leftarrow 1$  to  $w$  do // Warmup phase
4:   [...] // Do training as usual
5: for  $step \leftarrow w + 1$  to  $\infty$  do
6:    $x, y \leftarrow$  next training example
7:    $y' \leftarrow h_{\theta, \alpha + \alpha_0}(x)$ 
8:    $G \leftarrow g(y, y')$  // Get the performance measure
9:   if  $G > t_{\text{accept}}$  then
10:     $\alpha \leftarrow \alpha + i$  // We beat  $t_{\text{accept}}$ 
11:   else
12:     if  $G < t_{\text{reject}}$  then
13:        $\alpha \leftarrow \alpha * d$  // We fall below  $t_{\text{reject}}$ 
14:   [...] // Perform the rest of training as usual

```

that is, the way the decrement is done. As the name suggests, during a decrement, the current α is multiplied by the decrement d , which must be $0 \leq d < 1$. As stated before, we get this idea from the TCP’s congestion avoidance scheme [26].

4 Experiments and results

4.1 Experimental settings

We consider the problem of training small and medium sized networks in classification tasks on MNIST [27], FashionMNIST [28], CIFAR10 and CIFAR100 [29]. We applied minimal data augmentation, namely only horizontal flips (for

CIFAR10 and CIFAR100) and normalization of the input image. The trained architectures are both small fully connected networks and medium sized convnets. We used SGD as our optimizer with $\text{lr} = 0.01$, $\text{momentum} = 0.9$ and $\text{weight decay} = 0.001$. We used the AIMD noise scheduling algorithm with $\alpha_0 = 0.001$, $i = 0.0001$ and $d = 0.5$ as shown later. We used the training accuracy of the processed batch as the performance measure. In order to determine t_{accept} and t_{reject} , we first trained the baseline and measured its accuracy a on the test set, then we set t_{accept} and t_{reject} as $1.0 \cdot a$ and $0.8 \cdot a$ respectively. The idea behind this choice is that we can only increase α when the model is in the ballpark of a good solution. To achieve best results, we freeze the value of α for the last 8 epochs on a fixed noise level ($\alpha = 0.05$) during the training of the analog networks. This final fixed level global noise value will also be used as salt (see Section 4.2) for our adversarial resistance tests. Note that, unless otherwise stated, the α parameter is set to 0 at evaluation.

4.2 Results

We trained our models for 32 ($24 + 8$ for analog models) epochs (MNIST, FashionMNIST) and 64 ($56 + 8$ for analog models) epochs (CIFAR 10, CIFAR 100), picking the best test accuracy at every epoch end. In this section we present the salient results obtained by our proposed method against the baseline.

Do we really need α scheduling? We tested the performance of analog networks trained with a constant level of noise $\alpha = 0.05$ on CIFAR100 and $\alpha = 0.2$. We decided to test these value since they are, respectively, the fine tune values for our regularly scheduled analog models and the average noise level reached at the end of epoch 56 by our schedulers. This way, if the effect of scheduling is not relevant, we should expect negligible differences in the performances of non-scheduled versus scheduled models. Table 2 shows that, while injecting low noise levels without scheduling the intensity can still improve the results, scheduling further boosts the performances. Furthermore, we notice that the results are equivalent for both AIAD and AIMD. This is to be expected, as the decrease correction should (with a careful enough choice of the t_{accept} and t_{reject}) be executed the least possible and only serve as a guard against a pathological increment of the noise level, which may lead to decreased model capacity overall. During our tests, the differences in performances are negligible for the AIAD and AIMD schedulers. Thus, we will only report the results from the AIMD scheduler.

Accuracy Table 3 shows that the analog models are consistently able to outperform the baseline. This becomes more evident as the task is less “saturated”, that is, there is more room for improvement over the baseline.

Input perturbation resilience We tested the resistance of our model to additive input perturbations, multiplicative input perturbations and salt and pepper

Table 2. Accuracy of no scheduling vs AIAD vs AIMD scheduling on CIFAR100

	Fixed ($\alpha = .05$)	Fixed ($\alpha = .2$)	AIAD (ft. $\alpha = .05$)	AIMD (ft. $\alpha = .05$)
Const	0.5473	0.5429	0.5598	0.5598
Sqrt	0.5656	0.4859	0.5758	0.5758
Log	0.5493	0.5492	0.5695	0.5695
Mass	0.5682	0.4935	0.5760	0.5760

Table 3. Accuracy

	MNIST	FashionMNIST		CIFAR10	CIFAR100
MLP - Base	0.979	0.883	CNN - Base	0.7838	0.5367
MLP - Const	0.98	0.882	CNN - Const	0.8107	0.5606
MLP - Sqrt	0.98	0.882	CNN - Sqrt	0.8195	0.5751
MLP - Log	0.975	0.883	CNN - Log	0.8193	0.5693
MLP - Mass	0.978	0.884	CNN - Mass	0.819	0.5739

perturbations. The measured accuracy values are reported in table 4 and the perturbation level is indicated as p .

Table 4. Input perturbation accuracy for (from left to right) additive, multiplicative, salt and pepper perturbation on CIFAR10.

	$p = .05$	$p = .1$	$p = .2$		$p = .05$	$p = .1$	$p = .2$		$p = .05$	$p = .1$	$p = .2$
Base	0.668	0.348	0.190	Base	0.763	0.657	0.410	Base	0.373	0.224	0.135
Const	0.688	0.410	0.276	Const	0.791	0.676	0.454	Const	0.442	0.300	0.207
Sqrt	0.658	0.376	0.260	Sqrt	0.780	0.646	0.430	Sqrt	0.428	0.291	0.190
Log	0.706	0.436	0.247	Log	0.797	0.690	0.488	Log	0.457	0.289	0.163
Mass	0.656	0.374	0.251	Mass	0.780	0.646	0.432	Mass	0.424	0.276	0.205

We can see that generally, input perturbations do still affect the analog models, but at a less steep decrease in their performance (as p increases).

State perturbation resilience We tested the resistance of our model to state perturbations. In particular, we instantiated an analog model with “constant” heuristic with the weights of all our trained models, and tested accuracy as α increases. The results are reported in table 5.

We notice that state perturbation drastically affect the baseline performance, while being noticeably less effective against the analog models.

Table 5. State perturbation accuracy on CIFAR10.

	$\alpha = .05$	$\alpha = .15$	$\alpha = .3$
Base	0.7724	0.6393	0.3224
Const	0.8099	0.7868	0.6664
Sqrt	0.8167	0.7722	0.5707
Log	0.8152	0.7945	0.6619
Mass	0.8135	0.7690	0.5548

Resistance to adversarial attacks We tested the resistance against FSGM attacks of the CIFAR100 trained ConvNets. For our analog layer, we tried the adversarial attack both on the network without noise at test time (no salt) and with noise (unknown to the attacker, referred to as salt). Table 6 compares the resilience of the CNN trained on CIFAR10 and CIFAR100 against FSGM attacks. For the analog model, in particular, we consider two scenarios: in the first, the model is used with noise equal to zero, while in the second, the model is computed at a certain α level, with the resulting noise tensors $\beta_i \lambda$ (the whole of which we will refer to as “salt”) not known to the attacker. We discuss in section 5.4 why this attack scenario is reasonable. We measure improvements in adversarial resistance by using analog layers, further boosted by the usage of salt. The choice of salt level is a balancing choice: while more salt usually means a lower base performance, it also translates into higher adversarial resilience.

Table 6. Accuracy against FSGM attacks with various intensities of ϵ as reported on the top row on our CNN trained on CIFAR10 and CIFAR100

CIFAR10	$\epsilon = .01$	$\epsilon = .02$	$\epsilon = .03$	CIFAR100	$\epsilon = .01$	$\epsilon = .02$	$\epsilon = .03$
Base	0.1523	0.0189	0.0025	Base	0.0999	0.0215	0.0075
Const	0.2958	0.0617	0.01	Const	0.1393	0.0348	0.0116
Sqrt	0.3441	0.0828	0.02	Sqrt	0.2133	0.0715	0.0319
Log	0.3233	0.0773	0.0147	Log	0.165	0.045	0.0181
Mass	0.3731	0.1089	0.0288	Mass	0.2081	0.0675	0.029
Const + salt	0.3208	0.0719	0.0116	Const + salt	0.1526	0.0382	0.0131
Sqrt + salt	0.3870	0.1050	0.0268	Sqrt + salt	0.2325	0.0806	0.0363
Log + salt	0.3555	0.0864	0.0182	Log + salt	0.1825	0.0503	0.0194
Mass + salt	0.4110	0.1303	0.0347	Mass + salt	0.2288	0.0762	0.0335

4.3 Feature Maps Entropy

We measured the entropy of the feature maps of our CIFAR100 ConvNet. In particular, we measured the per-channel entropy after the Max Pooling operation

of each convolutional layer. To do this, we first normalized the channels in $[0-1]$ using the minimum and maximum value obtainable, per channel. Then, after binning in 256 bins, we measured the Shannon Entropy for each channel. The average entropy values are reported in Table 7. In general, analog models exhibit slightly less entropy across their channels.

Table 7. Average of per-channel entropy on our ConvNets trained on CIFAR10 and CIFAR100.

CIFAR10	Pool 1	Pool 2	Pool 3	CIFAR100	Pool 1	Pool 2	Pool 3
Base	7.2524	7.6412	7.4368	Base	7.3198	7.6784	7.4173
Const	7.2287	7.5294	7.4162	Const	7.2988	7.6164	7.3963
Sqrt	7.2176	7.5186	7.399	Sqrt	7.2738	7.5545	7.3723
Log	7.2246	7.5243	7.4066	Log	7.2886	7.5856	7.3881
Mass	7.2008	7.4957	7.4041	Mass	7.2522	7.5486	7.3717

5 Discussion

5.1 The analog framework as a form of curriculum learning

We can characterize our proposed framework as a curriculum learning technique. Curriculum learning is a paradigm rooted in how humans and animals seem to learn [34], where better results can be achieved by organizing training in a specific way. In particular, our method aims to vary the complexity of the task, which is dependant on the α parameter. As a result, our approach can learn more robust representations as compared to baseline models.

5.2 A regularization technique?

We can consider our proposed framework as a sort of regularization technique, as the additive noise on a certain layer output can become (depending on the activations) a multiplicative noise for the next layer, hence directly affecting weight gradient computation, which is in of itself regularizing. Moreover, our experiments show the proposed approach decreases the entropy of feature maps, which likely is an effect of the regularization induced by the injection of noise.

5.3 Is an analog model a consensus learner?

We speculate that the analog model is also a consensus based learner, in the sense that, at each forward pass of the training process (and inference if we are using salt), we are effectively training on `batch_size` samples of our stochastic model h_θ , which are close but not equal to the “expected” model (we better discuss

and formalize this concept in section 5.4). In particular, this may be useful in escaping or avoiding narrow local minima, as it becomes more and more difficult for the training to have most of these samples stuck in the same minima as the number of samples (which is equal to the batch size) increases. For this, we also suspect that analog layers may benefit for batch size scheduling.

5.4 Can We Defend an Open-Weights Model with... Salt?

Open-source models are a cornerstone in many AI applications. Often, large foundational models are released along with their architecture and a training checkpoint, eliminating the need for users to train a network from scratch, process often prohibitively expensive. While these open weights provide many individuals with access to powerful AI models, this also means that finding adversarial attacks against just one of these open-weight models can compromise the security of all services utilizing it. Although the results obtained with our method are not enough to claim increased adversarial resistance, they do suggest a potential strategy for developing public-weight, adversarial-resistant models.

In particular, let $h_{\theta, IV}$ represent a generic trained model, with θ being its public parameters and IV a set of stochastic parameters used for computation, that we call “salt”. In particular, let \mathcal{D} be the public joint distribution of the entire salt, that is, $IV \sim \mathcal{D}$. Note that the word “public” means that the value of θ and \mathcal{D} are known to the attacker. At runtime, the user samples IV from \mathcal{D} , effectively obtaining a certain instance of the model $h_{\theta, IV}$. Note that the sampling process is private, meaning that the value of IV is secret and not known by the attacker. Without further information, a reasonable approach for the adversary is to attack $h_{\theta, \mathbb{E}(\mathcal{D})}$.

If we can develop a method to train a “family” of models $h_{\theta, \cdot}$ such that it yields very similar results for different salt samples of \mathcal{D} while still maintaining sufficiently different internal states with regards to some permutation-invariant norm, we can potentially elude white-box adversarial attacks by obfuscating the internal details of the model’s operation behind the sampling of IV itself. This approach could serve as a robust defense mechanism against adversarial threats in open-weight models. The proposed analog layer allows to obtain models with the aforementioned properties, while the scheduling process allows to model \mathcal{D} as desired, for example by choosing a target α .

6 Conclusion

In this paper, we proposed and investigated the analog model, which, taking inspiration from analog circuits, provides a training framework that can boost performance and resilience of the trained models. We presented a definition of analog layer and analog model, and described two simple noise variation algorithms that can be used to effectively vary the noise level during training in an unsupervised way. We have measured the improvement over baselines on classification tasks in both Fully Connected networks and Convolutional Networks,

as well as perturbation resistance and adversarial attack resistance, and found that our method can be effectively used to improve these properties. Finally, starting from our findings on the Analog Layer, we discussed a possible more general approach to improve adversarial resistance in neural networks

Future work Future studies should consider the usage of other analog function, bigger datasets and state-of-the-art architectures, as well as different tasks and domains. Our scheduling algorithms, while simple and effective, are reliant on the chosen values of t_{accept} and t_{reject} . Furthermore, our proposed way of choosing said values requires training a baseline first, which may be costly for very big networks. Future works should consider ways to automatically find or schedule these values, or explore noise variation algorithms that don't need these kind of hard-coded thresholds altogether. Finally, to better assess the more advanced adversarial techniques such as the Carlini and Wagner's Attack [30], DeepFool [33] and even adaptive attacks [31]. We suspect that, due to the stochastic nature of our resulting model (when used with salt), iterative attacks may be the best fit, as demonstrated against SAP [32] in [31].

References

1. Jastrzbski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., Storkey, A. (2017). Three factors influencing minima in SGD. arXiv preprint [arXiv:1711.04623](https://arxiv.org/abs/1711.04623)
2. Smith, L. N. (2017, March). Cyclical learning rates for training neural networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 464-472). IEEE
3. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199)
4. Xu, H., Ma, Y., Liu, H.C., Deb, D., Liu, H., Tang, J.L., Jain, A.K.: Adversarial attacks and defenses in images, graphs and text: A review. *Int. J. Autom. Comput.* **17**, 151–178 (2020)
5. Goodfellow, I. J., Shlens, J., Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. arXiv preprint [arXiv:1412.6572](https://arxiv.org/abs/1412.6572)
6. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint [arXiv:1207.0580](https://arxiv.org/abs/1207.0580)
7. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 1097-1105)
8. Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., Shen, F. (2022). Image data augmentation for deep learning: A survey. arXiv preprint [arXiv:2204.08610](https://arxiv.org/abs/2204.08610)
9. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *Journal of big data* **6**(1), 1–48 (2019)
10. Antoniou, A., Storkey, A., Edwards, H. (2017). Data augmentation generative adversarial networks. arXiv preprint [arXiv:1711.04340](https://arxiv.org/abs/1711.04340)

11. Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., Haworth, A.: A review of medical image data augmentation techniques for deep learning applications. *J. Med. Imaging Radiat. Oncol.* **65**(5), 545–563 (2021)
12. Chen, H., Cao, P. (2019, July). Deep learning based data augmentation and classification for limited medical data learning. In 2019 IEEE international conference on power, intelligent computing and systems (ICPICS) (pp. 300-303). IEEE
13. Loshchilov, I., Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint [arXiv:1608.03983](https://arxiv.org/abs/1608.03983)
14. Smith, S. L., Kindermans, P. J., Ying, C., Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. arXiv preprint [arXiv:1711.00489](https://arxiv.org/abs/1711.00489)
15. Ba, J., Frey, B. (2013). Adaptive dropout for training deep neural networks. *Advances in neural information processing systems*, 26
16. Li, Z., Gong, B., Yang, T. (2016). Improved dropout for shallow and deep learning. *Advances in neural information processing systems*, 29
17. Achille, A., Soatto, S.: Information dropout: Learning optimal representations through noisy computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(12), 2897–2905 (2018)
18. Kingma, D. P., Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint [arXiv:1312.6114](https://arxiv.org/abs/1312.6114)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826)
20. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103)
21. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S. (2015, June). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning* (pp. 2256-2265). PMLR
22. Huang, G., Sun, Yu., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep Networks with Stochastic Depth. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9908, pp. 646–661. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_39
23. Han, B., Sim, J., Adam, H. (2017). Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3356-3365)
24. Noh, H., You, T., Mun, J., Han, B. (2017). Regularizing deep neural networks by noise: Its interpretation and optimization. *Advances in neural information processing systems*, 30
25. French, R.M.: Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **3**(4), 128–135 (1999)
26. Eddy, W., Ed., “Transmission Control Protocol (TCP)”, STD 7, RFC 9293, <https://doi.org/10.17487/RFC9293>, August 2022, <<https://www.rfc-editor.org/info/rfc9293>>
27. Lecun, Y., Cortes, C., Burges, C. J. (1998). Mnist. The MNIST Database of handwritten digits
28. Xiao, H., Rasul, K., Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint [arXiv:1708.07747](https://arxiv.org/abs/1708.07747)
29. Krizhevsky, A., Hinton, G. (2009). Learning multiple layers of features from tiny images
30. Carlini, N., Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 39-57). Ieee

31. Athalye, A., Carlini, N., Wagner, D. (2018, July). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International conference on machine learning (pp. 274-283). PMLR
32. Dhillon, G. S., Azizzadenesheli, K., Lipton, Z. C., Bernstein, J., Kossaifi, J., Khanna, A., Anandkumar, A. (2018). Stochastic activation pruning for robust adversarial defense. arXiv preprint [arXiv:1803.01442](https://arxiv.org/abs/1803.01442)
33. Moosavi-Dezfooli, S. M., Fawzi, A., Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2574-2582)
34. Bengio, Y., Louradour, J., Collobert, R., Weston, J. (2009, June). Curriculum learning. In Proceedings of the 26th annual international conference on machine learning (pp. 41-48)



Improving Out-of-Distribution Data Handling and Corruption Resistance via Modern Hopfield Networks

Saleh Sargolzaei and Luis Rueda^(✉)

School of Computer Science, University of Windsor, Windsor, Canada
lrueda@uwindsor.ca

Abstract. This study explores the potential of Modern Hopfield Networks (MHN) in improving the ability of computer vision models to handle out-of-distribution data. While current computer vision models can generalize to unseen samples from the same distribution, they are susceptible to minor perturbations such as blurring, which limits their effectiveness in real-world applications. We suggest integrating MHN into the baseline models to enhance their robustness. This integration can be implemented during the test time for any model and combined with any adversarial defense method. Our research shows that the proposed integration consistently improves model performance on the MNIST-C dataset, achieving a state-of-the-art increase of 13.84% in average corruption accuracy, a 57.49% decrease in mean Corruption Error (mCE), and a 60.61% decrease in relative mCE compared to the baseline model. Additionally, we investigate the capability of MHN to converge to the original non-corrupted data. Notably, our method does not require test-time adaptation or augmentation with corruptions, underscoring its practical viability for real-world deployment. (Source code publicly available at: <https://github.com/salehsargolzaee/Hopfield-integrated-test>)

Keywords: Modern Hopfield Networks · OOD Robustness · Computer Vision · Autoencoders · Convolutional Neural Networks

1 Introduction

The effectiveness of modern computer vision algorithms relies heavily on the assumption that data is independent and identically distributed (i.i.d.). Consequently, challenges arise in adapting these algorithms to generalize to out-of-distribution data in real-world scenarios, where visual corruption caused by adverse weather, lighting variations, and other factors is prevalent [19]. Studies have revealed a significant decrease in the generalization capability of models trained on clean data when exposed to these corruptions [6, 13].

In this study, we propose integrating a pre-trained Modern Hopfield Network (MHN) associative memory as an input module for the current models.

Associative memory is capable of recovering the original input based on partial information [7]. We suggest that adding this module, pre-trained to remove Gaussian noise from clean data, would help recover the original data in case of various corruptions. The proposal has advantages when compared with successful methods in combating corruption, including test-time adaptation (TTA) [22, 25] and domain adaptation (DA) [20].

TTA requires updating the model weights during the test time to adapt to corrupted data in test data. This adaptation process can introduce new challenges caused by batch size [14] or temporary traits of test data [22, 26]. However, our method does not require any test-time adaptation and can be trained offline and used during the test-time.

DA involves adjusting models that have been trained on one set of data (the source - in this case, clean images) so that they can be used on another set for which only unlabeled samples are available (the target - in this case, the corrupted images) [20]. This process is most effective when source and target domain data are available simultaneously. However, our method does not require access to the target corruptions, making it more suitable for real-world applications.

Notably, the proposed method can still be combined with the above-mentioned techniques. Our contributions can be summarized as follows:

- We propose a general pre-training scheme with Modern Hopfield Networks to build generic extensions that enhance test-time robustness against corruptions for any baseline model trained on a clean dataset.
- We develop a test-time integration algorithm using the pre-trained extension and validate its effectiveness on the MNIST-C dataset.
- We demonstrate the superiority of modern Hopfield networks in tolerating various types of corruption, beyond just noise, by comparing our extension with a convolutional denoising autoencoder pre-trained using the same scheme.
- We provide insights into the robustness of our integration algorithm when incorporating non-effective modules. We show that the algorithm maintains baseline performance even with an ineffective convolutional denoising autoencoder.
- We demonstrate the superiority of our algorithm compared to other offline methods designed to handle unseen data (corruptions). We also show that our method is comparable to test-time adaptive (TTA) methods and can be combined with TTA or offline methods to enhance robustness further.

2 Problem Statement

We focus on a new approach to address the issue of encountering out-of-distribution and corrupted data during testing. Adapting individual models to different types of corruption and changes in distribution can be time-consuming and repetitive. Our goal is to develop a memory layer capable of swiftly retrieving clean or sufficiently clean data from corrupted inputs in real time. If successful,

this layer can be integrated into any pre-trained classifier trained on clean data, thereby enhancing its robustness and adaptability.

Consider a classifier $f(x)$ trained on a dataset $\{(x_i, y_i)\}_i$, where $(x_i, y_i) \sim \mathcal{D}$. Let C be a set of corruption functions, and let $\mathbb{P}_C(c)$ represent the approximate frequency of corruption $c \in C$ in the real world. The task of corruption robustness can be defined as follows [6]:

$$\mathbb{E}_{c \sim \mathbb{P}_C} \left[\mathbb{P}_{(x,y) \sim \mathcal{D}} (f(c(x)) = y) \right], \quad (1)$$

where $\mathbb{P}_{(x,y) \sim \mathcal{D}} (f(c(x)) = y)$ is the probability that the classifier f correctly classifies the corrupted input $c(x)$. Our goal is to find an associative memory function h such that $(h(c(x_i)), y_i)$ is approximately distributed as \mathcal{D} . We consider this goal achieved if:

$$\mathbb{P}_{(x,y) \sim \mathcal{D}} (f(h(c(x))) = y) \approx \mathbb{P}_{(x,y) \sim \mathcal{D}} (f(x) = y), \quad (2)$$

where $\mathbb{P}_{(x,y) \sim \mathcal{D}} (f(x) = y)$ is the probability that the classifier f correctly classifies the original input x .

3 Modern Hopfield Networks

The classical Hopfield network was introduced as an associative memory model [7]. The model can be formalized as a system with N binary neurons where activity of the neurons at time t can be represented by a N -dimensional state vector $\boldsymbol{\sigma}^{(t)} = (\sigma_i^{(t)})_{i=1}^N$. In the original model, the states were considered to be binary, where $\sigma_i^{(t)} \in \{-1, +1\}$. In the classical Hopfield network, the update rule for each neuron is given by [7]:

$$\sigma_i^{(t+1)} = \text{Sign} \left[\sum_{j=1}^N T_{ij} \sigma_j^{(t)} \right] = \text{Sign} \left[\mathbf{T} \boldsymbol{\sigma}^{(t)} \right]_i, \quad (3)$$

where \mathbf{T} is a symmetric real-valued connection matrix with zeros on the main diagonal, specifying the pairwise connection strength among neurons. It can be shown that this update rule may result in a monotonic decrease of the following energy function:

$$E = - \sum_{i,j=1}^N \sigma_i T_{ij} \sigma_j = -\boldsymbol{\sigma}^T \mathbf{T} \boldsymbol{\sigma}. \quad (4)$$

Therefore, by following the update rule in Equation (3), the energy function may converge to a local minima. These local minima can be utilized to store memory (pattern), in such a way that by applying the update rule on a corrupted initialization of the memory, the original memory can be retrieved. The classic method for storing these memories involves encoding them in the weight matrix using the Hebb rule, which in its simplest form is:

$$T_{ij} = \sum_{\mu=1}^K \xi_i^\mu \xi_j^\mu, \quad (5)$$

where the set of vectors $\{\xi^\mu\}_{\mu=1}^K$ represent K patterns one wishes to store. In the classic Hopfield network, it has been shown that the maximal storage capacity in the case of random memories is in the order of $K \approx 0.14N$ [1, 12]. However, the network’s storage capacity can be increased by introducing non-linear functions to the energy, which may result in higher than quadratic interactions between the neurons [3, 8]. The following general form of energy function can characterize these Modern Hopfield Networks (MHN):

$$E = -h \left(\sum_{\mu=1}^K F(\sigma^T \xi^\mu) \right), \quad (6)$$

where $F(\cdot)$ represents a rapidly growing smooth function and $h(\cdot)$ is a strictly monotonic and differentiable function that can preserve the stability and locations of local minima. Setting $F(x) = e^x$ has been proved to result in a theoretical capacity of $K \approx e^{\alpha N}$, $\alpha < \frac{\ln(2)}{2}$, which is exponential in the number of neurons N [3]. We utilize the continuous state MHN introduced in [16]. The energy function of this model is obtained by setting $h(x) = \log(x)$ and $F(x) = e^{\beta x}$ in Equation (6), where β is a positive value. Due to the generalization to a continuous state vector, $\sigma \in \mathfrak{R}^N$, regularization terms were added to ensure that the energy is bounded and the norm of the state vector remains finite. The proposed energy function was expressed as follows:

$$E = -\beta^{-1} \log \left(\sum_{\mu=1}^K e^{\beta \sigma^T \xi^\mu} \right) + \frac{1}{2} \sigma^T \sigma + \beta^{-1} \log K + \frac{1}{2} M^2, \quad (7)$$

where $M = \max_\mu \|\xi^\mu\|$. The update rule for minimizing this energy is:

$$\sigma^{(t+1)} = \mathbf{X}_{\text{softmax}}(\beta \mathbf{X}^T \sigma^{(t)}), \quad (8)$$

where $\mathbf{X} = (\xi^1, \dots, \xi^K)$ forms a matrix by stacking memory vectors as its columns. In the following sections, we use a particular version of this model.

4 Methodology

We propose a two-step approach to enhance the model’s robustness against data corruption. First, we train a “HopfieldPooling” layer, a specialized variant of the continuous state Modern Hopfield Network (MHN) as introduced by Ramsauer et al. [16], on a denoising task. The primary objective is to develop an algorithm to integrate this trained HopfieldPooling module into an existing baseline model during the testing phase, thereby improving its resilience to various forms of data corruption. Both the baseline model and the HopfieldPooling layer are trained using clean data without access to future corrupted data during training.

4.1 Denoising Task

The denoising task serves as the preliminary step to achieve our main objective. Denoising tasks are widely used to learn robust data representations [23]. In this task, given an original input \mathbf{x} , we generate a corrupted version $\bar{\mathbf{x}}$ by adding Gaussian noise with mean zero and standard deviation 0.5. Consequently, $\bar{\mathbf{x}}$ can be modeled as a random variable following the distribution:

$$\bar{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}(\mathbf{x}, 0.5^2 \mathbf{I}). \quad (9)$$

We train the HopfieldPooling layer to minimize the mean squared error (squared L_2 -norm) between the original and denoised inputs, as defined by the following objective function:

$$\min_h \frac{1}{m} \sum_{i=1}^m \|\mathbf{x}^i - h(\bar{\mathbf{x}}^i)\|_2^2, \quad (10)$$

where $\{\mathbf{x}^i\}_{i=1}^m$ represents the set of training vectors and h denotes the Hopfield-Pooling layer.

4.2 Integration Algorithm

Once the HopfieldPooling layer is trained on the denoising task, we propose Algorithm 1 to integrate this module into an existing baseline model during the testing phase. The algorithm aims to process both corrupted data and the corrected version through the HopfieldPooling layer. It then makes predictions based on the input that generates more confidence for the most confident class. This integration aims to enhance the model’s ability to handle corrupted data effectively, leveraging the learned memory patterns from the HopfieldPooling layer to correct or mitigate the effects of corruptions encountered during testing.

5 Experiments

5.1 Experimental Setup

We conducted our experiments in the Google Colab environment using an NVIDIA Tesla T4 GPU with 15360 MiB of memory. We performed the training phase using Python 3.10.12 and PyTorch 2.3.0+cu121.

Dataset: For training purposes, we used 60,000 clear training samples from the MNIST dataset [9]. To test the proposed method, we utilized the MNIST-C dataset, a benchmark designed to evaluate the robustness of computer vision models [13]. This dataset was created by applying 15 types of corruptions to the 10,000 test images from the clean MNIST dataset, resulting in a total of 150,000 corrupted images. The corruptions include shot noise, impulse noise, glass blur, motion blur, shear, scale, rotate, brightness, translate, stripe, fog, spatter, dotted line, zigzag, and canny edges.

Algorithm 1 Test Model with Hopfield Integration.

```

1: Input:  $\mathcal{D}_{test} = \{\mathbf{x}^i\}_{i=1}^M$  ▷ Test dataset
2: Output:  $\mathcal{P} = \{p_i\}_{i=1}^M$  ▷ Predictions for test data
3:  $\mathcal{P} \leftarrow \emptyset$  ▷ Initialize the set of predictions
4: for all  $\mathbf{x}^i \in \mathcal{D}_{test}$  do
5:    $\mathbf{o}_f \leftarrow f(\mathbf{x}^i)$  ▷ Base model output (vector of class probabilities)
6:    $\mathbf{o}_h \leftarrow f(h(\mathbf{x}^i))$  ▷ Hopfield integrated output (vector of class probabilities)
7:    $(\max\_prob_f, \text{pred\_class}_f) \leftarrow (\max_j(\mathbf{o}_f[j]), \arg \max_j(\mathbf{o}_f[j]))$ 
8:    $(\max\_prob_h, \text{pred\_class}_h) \leftarrow (\max_j(\mathbf{o}_h[j]), \arg \max_j(\mathbf{o}_h[j]))$ 
9:   if  $\max\_prob_f > \max\_prob_h$  then
10:      $p_i \leftarrow \text{pred\_class}_f$  ▷ Choose base model prediction
11:   else
12:      $p_i \leftarrow \text{pred\_class}_h$  ▷ Choose Hopfield integrated prediction
13:   end if
14:    $\mathcal{P} \leftarrow \mathcal{P} \cup \{p_i\}$  ▷ Update test results with final prediction
15: end for
16: return  $\mathcal{P}$  ▷ Return all predictions

```

Implementation Details: To ensure repeatability, we trained and used the default convolutional neural network provided by the official PyTorch repository [15] without any modifications as the base model. This model definition is also employed in the benchmark paper of the MNIST-C dataset [13]. For training the HopfieldPooling layer, we used the hyperparameters shown in Table 1, which are inspired by the examples provided in the original publication’s repository [17]. For optimization, we adopted the AdamW optimizer introduced in [10], with the following default parameters: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\lambda = 0.01$. Our training process involves 20 epochs and a batch size of 20.

Table 1. Hyperparameters used for training the HopfieldPooling layer.

Hyperparameter	Value	Description
input_size	784	Dimension of the input vector $\bar{\mathbf{x}}$ in Equation (9)
hidden_size	8	Dimension of the association space N
num_heads	8	Number of parallel Hopfield heads
update_steps_max	5	Number of updates for each Hopfield head in each epoch
scaling	0.25	β parameter in Equation (8)

Evaluation Metrics: To evaluate the robustness gained by integrating the HopfieldPooling module into any baseline model, we use the mean Corruption Error (mCE) and relative mCE metrics, as established in benchmark publications [6, 13]. Given a corruption function $c \in \mathcal{C}$, a classifier g , and a baseline classifier f , we denote their error rates on c as E_c^g and E_c^f , respectively.

To account for varying corruption difficulties, we calculate the corruption error of g on c (CE_c^g) by normalizing its error rate with the baseline error rate:

$$CE_c^g = \frac{E_c^g}{E_c^f} \quad (11)$$

We also assess degradation relative to clean data (identity corruption i) by calculating the relative corruption error:

$$\text{relative } CE_c^g = \frac{E_c^g - E_i^g}{E_c^f - E_i^f} \quad (12)$$

From the two metrics defined above, we compute the mean values across all corruptions, resulting in mean CE (mCE) and relative mCE. Consequently, the mCE and relative mCE of the baseline model f are expected to be one (or 100%) due to these calculations. To quantify robustness improvements, we measure the reduction in mCE from 100%. Additionally, we calculate the gain in average corrupted accuracy.

5.2 Integration Results

Table 2 shows the evaluation metrics for the baseline model and Hopfield integration. The relative mCE and mCE are both 100% for the baseline model. With Hopfield integration, these values significantly decrease to 39.39% and 42.51%, respectively, demonstrating substantial improvements in robustness with reductions of 60.61 and 57.49 percentage points. Additionally, the average corruption accuracy increases from 75.92% for the baseline model to 89.76% with Hopfield integration, marking an improvement of 13.84 percentage points.

Table 2. Comparison of different evaluation metrics with and without Hopfield integration. Symbols \downarrow and \uparrow denote the desired direction of change for each metric.

Metric	Baseline	Hopfield-integrated	Improvement
Relative mCE (%) \downarrow	100	39.39	60.61
mCE (%) \downarrow	100	42.51	57.49
Average Corruption Accuracy (%) \uparrow	75.92	89.76	13.84

To better understand the effect of the HopfieldPooling layer on different corruptions, we compared the corrupted accuracy for each type. Table 3 displays these results. It can be observed that the corruption accuracy of the Hopfield-integrated model generally surpasses that of the baseline model. Specifically, the integration yields significant accuracy improvements of 82.65%, 50.13%, 43.42%, 18.41%, and 14.32% for fog, glass_blur, motion_blur, impulse_noise,

and brightness, respectively. Conversely, minor reductions or slight improvements in accuracy are observed for affine transformations, with changes of -3.06%, -0.94%, -0.68%, and 0.11% for translate, rotate, scale, and shear, respectively. These results highlight that the improvements are substantially more significant and robust, enhancing the baseline model’s performance under severe levels of corruption.

Table 3. Classification accuracy (%) of the baseline and Hopfield-integrated models on different types of corruptions.

Corruption	Baseline	Hopfield-integrated
identity (no corruption)	99.04	98.87
brightness	82.66	96.98
canny_edges	77.22	77.34
dotted_line	98.19	98.01
fog	14.35	97.00
glass_blur	39.66	89.79
impulse_noise	77.49	95.90
motion_blur	47.40	90.82
rotate	89.89	88.95
scale	89.10	88.42
shear	95.45	95.56
shot_noise	96.40	97.71
spatter	97.78	97.96
stripe	95.40	95.40
translate	47.42	44.36
zigzag	90.40	92.26
Average Corruption Accuracy	75.92	89.76

Our integration algorithm makes the final decision based on both the corrupted input and the input provided by the HopfieldPooling layer. We investigated how often the HopfieldPooling layer was used for each corruption and how it affected accuracy. The results are shown in Table 4. The Pearson correlation coefficient was found to be $r = 0.637$ with a p -value of 0.008. The coefficient suggests a moderate positive correlation between the usage of the HopfieldPooling layer and accuracy improvement. The p -value is well below the commonly accepted significance threshold of 0.05, indicating that this correlation is statistically significant. This suggests that higher usage of the HopfieldPooling layer is associated with greater improvements in accuracy.

Table 4 also illustrates that the most significant improvements are obtained by utilizing the HopfieldPooling layer for almost all the decisions. For instance,

in the case of fog and motion_blur, the proposed integration utilized the HopfieldPooling layer for 99.94% and 99.43% of the decisions, respectively. Also, it turns out that, even in the case of minor or no improvement in accuracy, as in shot_noise or shear, the baseline model utilized the HopfieldPooling layer for its final decisions. This behavior suggests that the Hopfield module also improves the final probabilities, and hence, the model’s confidence in selecting the correct class.

Table 4. HopfieldPooling usage and increase in accuracy for different corruptions.

Corruption	HopfieldPooling Usage (%)	Increase in Accuracy (%)
identity (no corruption)	20.47	-0.17
brightness	99.84	14.32
canny_edges	0.65	0.12
dotted_line	36.68	-0.18
fog	99.94	82.65
glass_blur	97.69	50.13
impulse_noise	86.36	18.41
motion_blur	99.43	43.42
rotate	72.68	-0.94
scale	49.36	-0.68
shear	74.56	0.11
shot_noise	82.81	1.31
spatter	50.55	0.18
stripe	0.00	0.00
translate	39.15	-3.06
zigzag	33.15	1.86
	Pearson Correlation (r)	p-value
	0.637	0.008

5.3 Ablation Study

To assess the potential of achieving similar results by integrating a different pre-trained denoising model, we replaced the HopfieldPooling layer with a stacked Convolutional Denoising Autoencoder (CDAE) [11, 24]. This ablation study is vital for determining whether the improvements observed are specifically attributed to the embedded memories of the Hopfield associative memory or if similar results can be reproduced using alternative techniques, which ultimately would lead to constructing a robust latent representation of input data.

Convolutional Autoencoder Implementation: Tables 5 and 6 provide the layers of the encoder and decoder parts of the convolutional autoencoder, respectively. After each 2D convolution and transposed convolution operation, a ReLU activation function is used, which is defined as $ReLU(x) = \max(0, x)$. Since the original input values are between 0 and 1, the final output is passed through a Sigmoid activation function, which reduces the output logits to values between 0 and 1, and which is defined as $Sigmoid(x) = \frac{1}{1+e^{-x}}$. We keep the experimental setup constant (cf. Section 5.1).

Table 5. Architecture of the encoder module.

Layer	Operation	Number of Kernels	Kernel Size	Stride	Padding	Output Shape
1	2D convolution	32	(3, 3)	1	1	(28, 28, 32)
2	2D max pooling-		(2, 2)	2	0	(14, 14, 32)
3	2D convolution	16	(3, 3)	1	1	(14, 14, 16)
4	2D max pooling-		(2, 2)	2	0	(7, 7, 16)
5	2D convolution	8	(3, 3)	1	1	(7, 7, 8)
6	2D max pooling-		(2, 2)	2	0	(3, 3, 8)

Table 6. Architecture of the decoder module.

Layer	Operation	Number of Kernels	Kernel Size	stride	padding	Output Shape
1	2D transposed convolution	8	(3, 3)	2	0	(7, 7, 8)
2	2D transposed convolution	16	(2, 2)	2	0	(14, 14, 16)
3	2D transposed convolution	32	(2, 2)	2	0	(28, 28, 32)
4	2D convolution	1	(3, 3)	1	1	(28, 28, 1)

Comparison on the Denoising Task: Fig. 1 illustrates a comparison of the mean squared error (MSE) per epoch on the preliminary denoising task (cf. Section 4.1) for the HopfieldPooling layer and CDAE. We observe that HopfieldPooling exhibits significant superiority in terms of MSE. HopfieldPooling reaches a training error of 0.016 at the second training epoch, while CDAE plateaus near 0.022 even after 20 training epochs.

Comparison on the Integration Algorithm: Fig. 2 illustrates the robustness metrics across three conditions: baseline model, CDAE integration, and HopfieldPooling integration. The metrics indicate that CDAE integration offers minimal improvements. Furthermore, Pearson correlation analysis between the

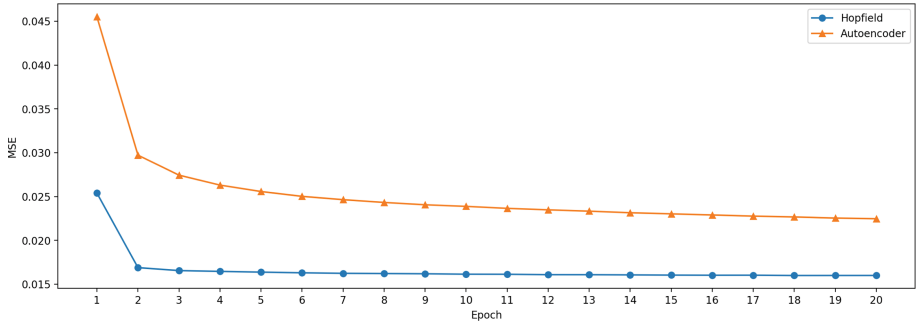


Fig. 1. Comparison of the mean squared error of different models per epoch during training on the denoising task.

number of decisions based on CDAE outputs and corrupted accuracy improvements yielded a correlation coefficient (r) of 0.427 with a p -value of 0.099. Since the p -value exceeds the commonly accepted significance threshold of 0.05, this correlation is not statistically significant. Consequently, we cannot assert a meaningful linear relationship between decisions based on CDAE output and improvements in corrupted accuracy.

Comparison of Corruption Removal: To investigate the challenges of CDAE in enhancing robustness compared to the HopfieldPooling layer, we analyzed the output of each pre-trained network when subjected to corrupted inputs. We present the outputs for both types of corruption where HopfieldPooling exhibited significant accuracy improvements (Fig. 3) and the affine transformations where the metrics showed limited improvements (Fig. 4).

The illustrations demonstrate that CDAE not only fails to remove corruption but also disrupts the digit pattern entirely. In some cases, these disruptions make classification extremely difficult, even for humans. This failure could be due to the inability of the latent representation to generalize beyond Gaussian noise corruptions. Yet, these results underscore the robustness of our proposed integration algorithm to ineffective modules. As previously shown in Fig. 2, the integration algorithm still caused slight improvements by adding CDAE, suggesting that the algorithm mainly decides based on the best input.

Conversely, the HopfieldPooling layer effectively mitigates most corruption. Notably, in the case of affine transformations shown in Fig. 4, the model successfully reconstructs the digits despite not being trained on any affine transformations. These results suggest that the integration algorithm’s inability to enhance robustness in affine transformations may be attributed to the base model’s limitations. Previous studies have also indicated that convolutional neural networks (CNNs) are vulnerable to simple transformations such as translation and rotation [4]. Nevertheless, the HopfieldPooling layer effectively generalizes to these transformations and reconstructs the correct digit. Additionally, it is observed

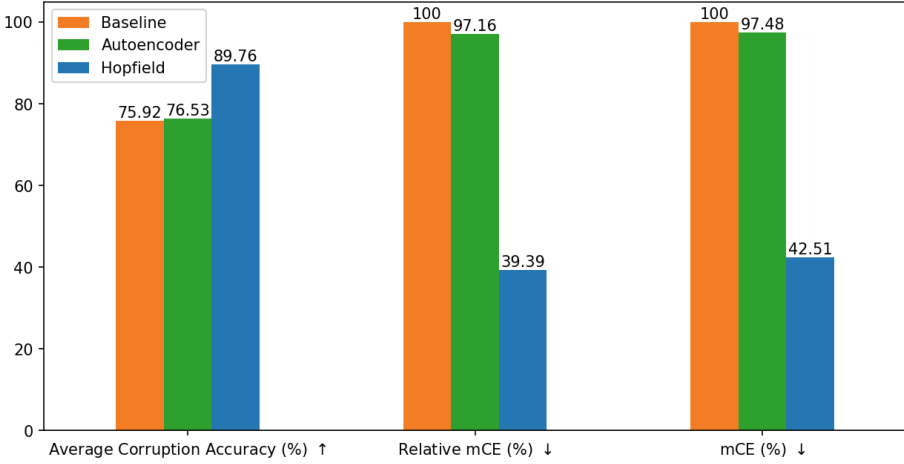


Fig. 2. Comparison of baseline, autoencoder-integrated, and Hopfield-integrated models on different robustness metrics during test time. Symbols ↓ and ↑ denote each metric’s desired direction of change.

that the digits become bolder, which may explain why the module increases the model’s confidence in many cases.

Corruption	fog		glass_blur		motion_blur		brightness	
True label	1	6	5	6	9	8	6	9
Corrupted Input								
HopfieldPooling Layer Output								
CDAE Output								

Fig. 3. The output of the HopfieldPooling layer and CDAE for corruptions on which notable improvement in robustness metrics is achieved with HopfieldPooling.

Corruption	translate		rotate		scale		shear	
True label	8	3	6	3	1	8	3	5
Corrupted Input								
HopfieldPooling Layer Output								
CDAE Output								

Fig. 4. The output of the HopfieldPooling layer and CDAE for corruptions made with affine transformations.

6 Discussion and Future Directions

In this section, we situate our method within the current state-of-the-art research on robustness to data corruption and outline potential future research opportunities by integrating these methods. Table 7 presents accuracy improvements achieved by training state-of-the-art computer vision models with various data augmentation techniques. Our method demonstrates superior performance across these methods. Additionally, we propose the HopfieldPooling layer as a versatile extension applicable to any baseline model. These augmentation techniques can be leveraged in future work to enhance the robustness of baseline models or to refine the training procedures for the Hopfield module. Furthermore, adversarial training methods could be employed in the training phase of the HopfieldPooling layer to develop a more generalized module resilient to a broader spectrum of corruptions [21].

Table 7. Comparison of accuracy improvements (%) using data augmentation methods and our method.

Method	Corrupted Accuracy Improvement \uparrow
Augment training data with 31 corruptions [13]	6.39
Tuned training with additive Gaussian and Speckle noise [18]	5.5
Augment training data with α - stable noise [28]	8.85
Augment training data using multi-scale random convolutions [27]	3.42
Hopfield integration method (ours)	13.84

Another promising direction is to incorporate our integration module into test-time adaptation (TTA) methods. Our method has already outperformed some of the leading TTA models, such as LAME [2]. For instance, Gong et al. [5]

reported a classification error of 11.8 on the MNIST-C dataset using the LAME method, whereas our method achieved a classification error of 10.24 without any adaptation. Despite this, their TTA method (NOTE) reduced the classification error to 7.1. We propose further investigations into applying different TTA methods to adapt the general HopfieldPooling layer during test time. The adapted layer can then be used for more than one baseline model.

Finally, continuous-state modern Hopfield networks [16] have gained significant attention in recent years. Our method exemplifies their application, yet numerous possibilities remain for future research. These include testing on color images, comparing various architectures and energy functions, and exploring the impact of hyperparameters on the Hopfield module.

7 Conclusion

Our study tackles the challenge of enhancing the reliability of computer vision models, particularly under test-time corruption. We introduce a universal integration algorithm that leverages a pre-trained modern Hopfield network on a clean dataset to significantly boost the performance of any baseline model. Our approach demonstrates comparable results to methods relying on extensive data augmentation or test-time adaptation, as evidenced by our experiments on the MNIST-C dataset. Moreover, our method’s versatility allows for seamless integration with other techniques. The critical role of the Hopfield network was highlighted by our comparison with a convolutional denoising autoencoder, which did not yield significant improvements, further validating the effectiveness of our proposed approach.

Acknowledgements. This research work has been partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC, and the Vector Institute for Artificial Intelligence.

References

1. Abu-Mostafa, Y., Jacques, J.S.: Information capacity of the hopfield model. *IEEE Trans. Inf. Theory* **31**(4), 461–464 (1985)
2. Boudiaf, M., Mueller, R., Ben Ayed, I., Bertinetto, L.: Parameter-free online test-time adaptation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8344–8353 (2022)
3. Demircigil, M., Heusel, J., Löwe, M., Uppang, S., Vermet, F.: On a model of associative memory with huge storage capacity. *Journal of Statistical Physics* **168**(2), 288–299 (May 2017), <http://dx.doi.org/10.1007/s10955-017-1806-y>
4. Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A.: A rotation and a translation suffice: Fooling cnns with simple transformations (2017)
5. Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., Lee, S.J.: Note: Robust continual test-time adaptation against temporal correlation (2023), <https://arxiv.org/abs/2208.05117>

6. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations (2019), <https://arxiv.org/abs/1903.12261>
7. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* **79**(8), 2554–2558 (1982), <https://doi.org/10.1073/pnas.79.8.2554>
8. Krotov, D., Hopfield, J.J.: Dense associative memory for pattern recognition (2016), <https://doi.org/10.48550/arXiv.1606.01164>
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
11. Masci, J., Meier, U., Cireřan, D., Schmidhuber, J.: Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) *ICANN 2011*. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_7
12. McEliece, R., Posner, E., Rodemich, E., Venkatesh, S.: The capacity of the hopfield associative memory. *IEEE Trans. Inf. Theory* **33**(4), 461–482 (1987)
13. Mu, N., Gilmer, J.: Mnist-c: A robustness benchmark for computer vision (2019), <https://arxiv.org/abs/1906.02337>
14. Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M.: Towards stable test-time adaptation in dynamic wild world (2023), <https://arxiv.org/abs/2302.12400>
15. PyTorch Contributors: Pytorch mnist example. <https://github.com/pytorch/examples/blob/main/mnist/main.py> (2024), accessed: 2024-06-29
16. Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G.K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., Hochreiter, S.: Hopfield networks is all you need (2021)
17. Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G.K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., Hochreiter, S.: Hopfield layers: Official implementation. <https://github.com/ml-jku/hopfield-layers> (2024), accessed: 2024-06-29
18. Rusak, E., Schott, L., Zimmermann, R., Bitterwolf, J., Bringmann, O., Bethge, M., Brendel, W.: Increasing the robustness of dnns against image corruptions by playing the game of noise (2020)
19. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision* **126**(9), 973–992 (Mar 2018), <http://dx.doi.org/10.1007/s11263-018-1072-8>
20. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation (2020), <https://arxiv.org/abs/2006.16971>
21. Stutz, D., Hein, M., Schiele, B.: Confidence-calibrated adversarial training: Generalizing to unseen attacks. In: *International Conference on Machine Learning*. pp. 9155–9166. PMLR (2020)
22. Su, Y., Xu, X., Jia, K.: Towards real-world test-time adaptation: Tri-net self-training with balanced normalization (2023), <https://arxiv.org/abs/2309.14949>
23. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on Machine learning*. pp. 1096–1103 (2008)
24. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., Bottou, L.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* **11**(12) (2010)

25. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization (2021), <https://arxiv.org/abs/2006.10726>
26. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. *Neurocomputing* **312**, 135–153 (Oct 2018), <http://dx.doi.org/10.1016/j.neucom.2018.05.083>
27. Xu, Z., Liu, D., Yang, J., Raffel, C., Niethammer, M.: Robust and generalizable visual representation learning via random convolutions. arXiv preprint [arXiv:2007.13003](https://arxiv.org/abs/2007.13003) (2020)
28. Yuan, X., Li, J., Kuruoğlu, E.E.: Robustness enhancement in neural networks with alpha-stable training noise (2023), <https://arxiv.org/abs/2311.10803>



A-FSL: Adaptive Few-Shot Learning via Task-Driven Context Aggregation and Attentive Feature Refinement

Riti Paul^(✉), Sahil Vora, Nupur Thakur, and Baoxin Li

School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281, USA

{rpaul12,svora7,nsthaku1,baoxin.li}@asu.edu

Abstract. Learning new categories with limited training samples presents a significant challenge for conventional deep learning frameworks. The few-shot learning (FSL) paradigm emerges as a potential solution to address practical constraints in this challenge. The primary difficulties in FSL are insufficient prior knowledge and ineffective alignment of clusters to their corresponding classification vectors in the pre-trained feature space. While many FSL methods employ task-agnostic instances and class-specific embedding functions, we argue that incorporating task-specific knowledge is crucial for overcoming FSL challenges. To achieve adaptability in FSL, we propose an Adaptive Few-Shot Learning (A-FSL) framework which (1) aggregates task-specific knowledge and adapts the classification vectors in the pretrained feature space and (2) develops a query class correlation attention module to enhance cluster formation. By considering task-specific information at multiple scales of visual features, we can overcome the limitations of a fixed feature space and refine it to adapt classification and query vectors effectively. The A-FSL framework leads to well-formed clusters for novel classes where classification vectors are drawn toward the clusters, even in the 1-shot setting. Through comprehensive experimental evaluation, we show that our method outperforms the current state-of-the-art on benchmark datasets.

Keywords: Few-shot learning · Image Classification · Label Generalization

1 Introduction

In recent years, deep neural networks (DNNs) [13, 27, 30] have made impressive strides in many computer vision tasks such as object classification, scene classification, etc. Training a DNN requires much effort for data labeling and (typically) a gradient-based learning algorithm for inferring optimal model parameters based on the data. All these are generally expensive tasks in terms of

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_7.

manual or computational costs. Additionally, deep networks suffer from generalization issues and can only recognize classes in which they have been trained. In a real-world scenario, there is a possibility of encountering images of novel classes that the trained network has never seen before. In such cases, the deep network has to go through the computationally expensive training process again on the novel classes with sufficient data (to avoid overfitting), which requires extensive manual effort and previous label data to prevent catastrophic forgetting. This completely contrasts the cognitive abilities of humans to remember and recognize objects even in situations with limited available data. Mimicking such dynamic and efficient behavior of humans on an artificial system is a task known as Few-Shot Learning (FSL), which is a complex research challenge that offers numerous practical advantages.

FSL involves identifying new categories, even with just one or a few labeled examples, by utilizing the visual patterns learned from the base/known categories. Recent years have witnessed a surge of research interest in various FSL techniques [15, 16, 20, 32], which aim to decrease the reliance on labeled data in deep model training.

The FSL paradigm presents two main challenges: first, dealing with insufficient or noisy estimates due to limited labeled samples, and second, addressing the improper alignment of class-specific samples with classification vectors. To tackle the first challenge, various weight generation frameworks have emerged, employing meta-learning-based training strategies to transfer knowledge from similar base classes during training. Dynamic-FSL [6] is an early exploration in this direction. Additionally, incorporating semantic information has been explored as another avenue to enhance prior knowledge about a given class and its samples. Works like AM3 [37], SEGA [39], and LPE-FSL [40] contribute to this aspect. To circumvent the limited effectiveness of a fixed feature space during the transfer to unseen classes, some works such as LEO [25], DeepEMD [41], TADAM [19], AWGIM[7] adopt adaptive techniques to transform the pretrained feature space into task-specific representations and have shown promising performance in this direction.

A unified solution to address existing challenges involves integrating task-specific knowledge and adapting the representations based on the available task-specific relationship among samples. In certain scenarios where the classes within a task have higher inter-class similarity than their base classes, current weight generation frameworks often overlook the available task-specific prior information. Since the success of few-shot learning hinges on increased prior information availability, we propose **Adaptive Few-Shot Learning (A-FSL)**, a constrained task context aggregation and task-specific refinement strategy. This strategy updates and aligns the classifier weights more closely to the test samples while leveraging the readily available task information, both global and class-specific. Following the findings of [23], we propose to refine the pretrained feature space with class-specific spatial attention derived from task-specific context, which assists in adapting classifier weights and the query samples' representation and aligning them closer. Since the refinement procedure (regardless of whether it

is a support or a query instance) depends on task-specific contextual information, consistency is maintained across classes in the refined feature space. Furthermore, generating attention masks based on sample-specific correlation with available task-specific class feature maps facilitates the highlighting of class-relevant features in each sample, thereby localizing target objects and fostering a more discriminative feature space. This approach proves particularly effective in one-shot scenarios where the availability of labeled samples is limited, and the estimates may be noisy. In this work, our contribution can be summarized as follows:

- We explore the perspective of extracting task-specific contextual representation to adapt pretrained features consistently. By focusing on the unique relationship between samples in a task, we can adaptively extract information and enhance the discriminative power of the features. We propose to transfer prior information from samples available in a task based on a task-specific context.
- We propose a (Query) Class Correlation Attention Module - (Q)CCAM capable of highlighting class-relevant features for any given sample. Our method combines weight-generating and metric learning concepts to create an effective few-shot classification system.
- Experiments show the effectiveness of the proposed method, especially in the 1-shot setting.

2 Related work

In this section, we provide a brief discussion of the relevant existing works. The metric-learning based few-shot methods learn an embedding space where the images have similar embedding if they belong to the same class while the images belonging to different classes have different embeddings [5, 18, 36, 41]. As deep learning started gaining popularity, many neural-network-based metric-learning techniques were proposed. Prototypical Networks [28] use the mean of class support samples’ embeddings to generate the class prototypes. The nearest neighbor search is used for predictions of novel classes. [32] proposed Matching Networks, which utilize an attention mechanism to compare the test images from novel classes with the support samples using context knowledge. [29] proposed a Relation Network that uses a deep neural network to learn a transferable deep metric.

Similar to metric-learning methods, the parameter generation methods also use feature embeddings, but the way of using them is different. Parameter generation methods use these embeddings to predict the classification weights for novel classes. To eliminate the norm issue, the cosine classifier is a common choice for such methods [10, 21]. [39] proposed a mechanism to enhance the use of prior knowledge and use semantic knowledge along with visual features. [20] proposed an architecture called KTN, which combined visual feature learning, inferring knowledge, and predicting the classifier weights. [6, 21, 22] are other recent methods proposed for novel class weight generation. The parameter generation and the metric-learning-based methods are sometimes termed *lazy learning methods*.

Meta-learning-based methods are also a prominent genre in few-shot learning techniques. In meta-learning-based methods, the training data (base classes) is used to learn the meta-knowledge over a distribution of tasks, which teaches the network how to adapt to novel classes [1, 26]. There are various ways to distill the knowledge learned from the known classes, like learning a class-agnostic transformation using a regression network [34], using operations like temporal convolution and attention to collect the information from the past experiences [17], etc. With the rapid advancement in large vision-language models, prompt learning [11, 12] has emerged as an effective learning paradigm. However, we exclude these works from comparison due to significant architectural, training, and evaluation setup differences.

Our proposed method falls under metric learning, and following previous works, we use the episodic strategy of training and testing. This enhances the knowledge learned by the network by learning common and task-specific information, in turn increasing the adaptability of the network to unseen novel classes.

3 Proposed Method

3.1 Problem Definition

In the few-shot learning paradigm, the standard configuration involves a base dataset denoted as $D_{base} = \{x_i, y_i\}, \forall y_i \in Y_b$, which is an extensive labeled dataset, and x_i represents the pretrained feature. This dataset is pivotal for acquiring prior knowledge that can be transferred to novel labels $Y_n \in D_{novel}$. Few-shot learning is broadly defined as $Y_b \cap Y_n = \emptyset$, encompassing various variations. This study specifically addresses scenarios concerning mutually exclusive label sets originating from the same domain. In the novel dataset D_{novel} , each label is limited to K labeled samples, constituting the support set for training. For a label set featuring N novel labels, the few-shot task is expressed as an N -way K -shot problem. Additionally, semantic information in the form of $S = \{s_c\}$, where $c \in Y_b \cup Y_n$, is available. The primary objective is to learn a mapping $f : X \rightarrow Y_n$.

3.2 Task-Specific Classifier Weight Generator

This section presents our approach to generating classification weight vectors for novel classes. We leverage three sources of information within the fixed pretrained feature space to construct a classification vector. Drawing inspiration from prior work [6, 39], we incorporate class-specific samples (scaled-mean prototype), transfer knowledge from base classes (base prior transfer), and task-specific out-of-class samples (proposed task-specific prior transfer). Additionally, we refine the classifier weights and query vectors using our proposed $CCAM(\cdot)$ and $QCCAM(\cdot)$ modules, respectively.

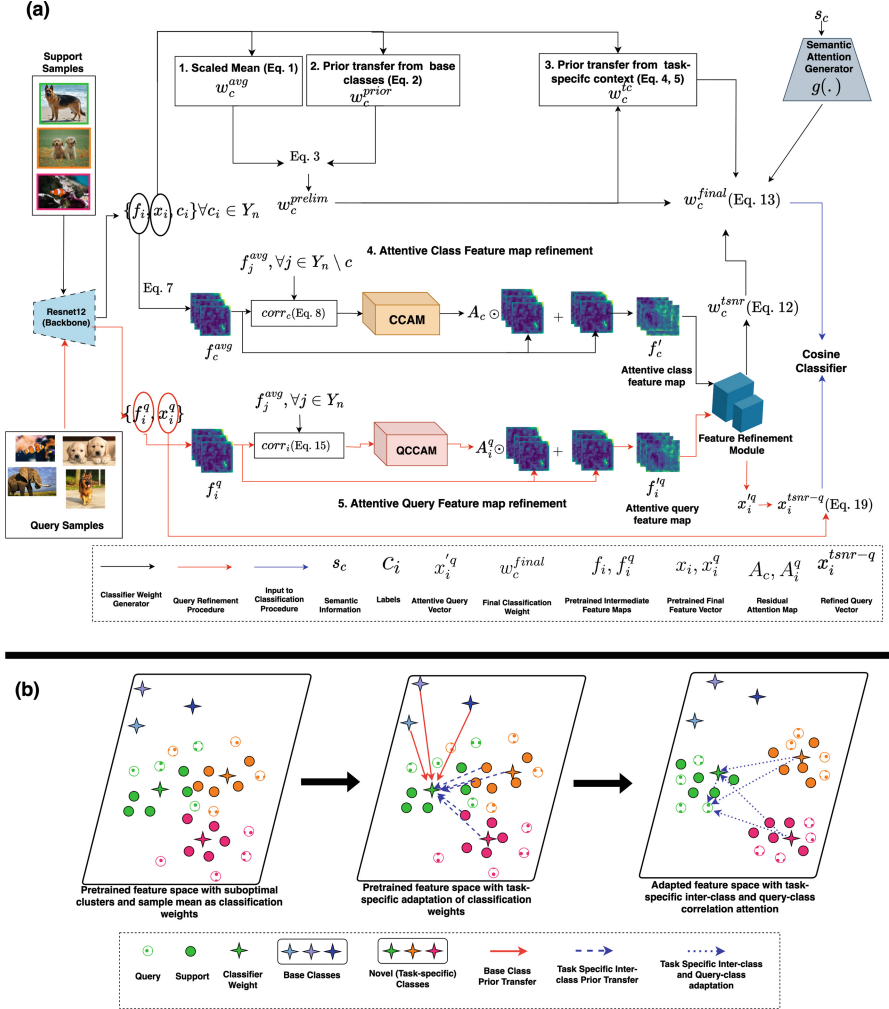


Fig. 1. (a) Framework architecture diagram for the proposed network. We propose Modules 3, 4, and 5. Modules 1, 2 and $g(\cdot)$ are adopted from the baselines [6, 39]. During stage 2 training, all the modules are trained except the backbone (Resnet-12). (b) Schematic illustration of our proposed strategy. Prior transfer from the base classes and task-specific inter-class relationships adapts the initial prototypes in the pretrained space. However, it still suffers from loosely formed clusters. Adapting the pretrained feature space with inter-class correlation and, most importantly, query-class correlation assists in localizing target features and well-formed clusters.

Scaled Mean Prototype: As shown in Prototypical Network [28], the classifier weight is equivalent to the mean of available support samples. Hence, we follow

the ideology of most metric-based methods and get the initial visual prototype:

$$w_c^{avg} = \frac{1}{|D_S^c|} \sum_{i=1}^{|D_S^c|} x_i, \forall x_i \in D_S^c \quad (1)$$

where D_S^c is the support set for class c for any given task T and $|D_S^c| = K$.

Prior transfer from base classes Y_b : Without any doubt, prototype generation in Equation 1 is highly unstable and noisy (especially in the 1-shot setting). To utilize further prior knowledge that is transferable from the available base class weights $W_{base} = \{w_c\}_{c=1}^{|Y_b|}$, we follow the work of Dynamic FSL [6] to transfer visual prior from base classification weights based on the cosine similarity:

$$w_c^{prior} = \frac{1}{|D_S^c|} \sum_{(x_i, y_i) \in D_S^c} \sum_{j \in Y_b} Att(\phi_q x_i, k_j) \cdot w_j \quad (2)$$

where $\phi_q \in \mathbb{R}^{d \times d}$ is a learnable weight matrix that transforms any given feature x_i to a query vector and k_j is a set of learnable key vectors corresponding to base classification weights w_j . $Att(\cdot, \cdot)$ is a cosine-based attention kernel that computes the similarity of the transformed feature and base class keys to transfer visual prior from base classes to novel class c . Combining 1 and 2, we model a preliminary visual prototype as follows:

$$w_c^{prelim} = \lambda_{avg} \times w_c^{avg} + \lambda_{prior} \times w_c^{prior} \quad (3)$$

where λ_{avg} and λ_{prior} are learnable scalar coefficients that control the contribution of each of the components. Next, we incorporate the task-specific information from available samples in the task and update the preliminary prototype.

Prior transfer from task-specific context: Aggregating inter-class relationships within a task is a valuable strategy for improving classification performance, especially when the similarity between task-specific classes surpasses that of the base classes. In such scenarios, utilizing task-specific prior information becomes crucial for refining classifier weights effectively. For example, in species classification, where certain breeds of dogs share greater similarities than others, incorporating inter-class relationships enhances the classifier’s ability to differentiate between visually and semantically similar classes. To leverage such knowledge, we provide the following learning strategy:

For every preliminary class prototype w_c^{prelim} (Equation 3), we define an information set $I^c = \{Z_c, Z'_c, W_{Y_n \setminus c}^{prelim}\}$, where Z_c represents a set of m closest out-of-class samples (from remaining $(N - 1)$ labels $\in Y_n \setminus c$), Z'_c contains the remaining $((N - 1) * K - m)$ out-of-class samples in a N -way K -shot task and $W_{Y_n \setminus c}^{prelim}$ represents the set of other preliminary class prototypes for task T . We compute feature similarity between these sets for each preliminary prototype to capture expressive correlations and aggregate them to create our task contextual information based on inter-class relationships, which is formed using the following protocol:

- Calculate distance between preliminary class prototype w_c^{prelim} and its corresponding m closest samples ($\in Z_c$) to create a m dimensional vector,

$$D_Z = [dist(w_c^{prelim}, Z_1), \dots, dist(w_c^{prelim}, Z_m)] \in \mathbb{R}^m$$

We leverage the correlation of m closest out-of-class samples with any prototype w_c^{prelim} to formulate task-specific context. Aggregating the individual correlation of the closest negative samples and other task-specific relationships (listed below) will assist in adapting the preliminary prototype with discriminative information.

- Compute the average distance between preliminary class prototype w_c^{prelim} and remaining out-of-class samples belonging to Z'_c ,

$$D_{Z'} = \frac{\sum_{i=1}^{K*(N-1)-m} dist(w_c^{prelim}, Z'_i)}{K(N-1) - m} \in \mathbb{R}^1$$

Farther samples have a higher probability of degrading the classifier quality. Therefore, we consider the average similarity of such samples.

- The relationship between preliminary class prototypes provides context regarding how much we can utilize out-of-class samples to update prototype w_c^{prelim} . Therefore, we consider the distance between preliminary class prototype w_c^{prelim} and other preliminary prototypes (individually) to create a $(N-1)$ dimensional distance vector, $D_{W_{Y_n \setminus c}^{prelim}} = [dist(w_c^{prelim}, W_{Y_n \setminus c(j)}^{prelim})], \forall j \in Y_n \setminus c$.

where $dist(.,.)$ denotes cosine similarity and $[.]$ is a concatenation operator. Also, $D_{W_{Y_n \setminus c}^{prelim}} \in \mathbb{R}^{N-1}$. We create our task context by concatenating these three sources of task-specific information, which is denoted as follows:

$$D_c^{tc} = [D_Z, D_{Z'}, D_{W_{Y_n \setminus c}^{prelim}}] \in \mathbb{R}^{m+N} \quad (4)$$

To transfer task-specific visual prior from the samples in Z_c , we use a similar network as defined in Equation 2. The cosine similarity-based task-specific transfer can be represented as:

$$w_c^{tc} = \frac{1}{|Z_c|} \sum_{i=1}^{|Z_c|} Att(\phi_{tc} D_c^{tc}, k_i) \cdot Z_c^i \quad (5)$$

where $\phi_{tc} \in \mathbb{R}^{(m+N) \times d}$ is a learnable weight matrix that transforms any task context D_c^{tc} for class c to a query vector and k_i is a set of learnable key vectors corresponding to samples in Z_c . $Att(.,.)$ is a cosine-based attention kernel that computes the similarity of the transformed feature and m out-of-class samples' keys to transfer visual prior from task samples to novel class c . We model our task-specific prototype in the pretrained feature space as the following:

$$w_c^{pt} = \lambda_{avg} \times w_c^{avg} + \lambda_{prior} \times w_c^{prior} + \lambda_{tc} \times w_c^{tc} \quad (6)$$

where λ_{avg} , λ_{prior} and λ_{tc} are learnable scalar coefficients.

Attentive Class Feature map refinement: However, as noted in [9], the effectiveness of weight-generation methods is restricted due to fixed pretrained feature space. To address this limitation, we use the concept of visual feature correlation to enhance discriminative features for support and query samples. The feature correlation map is treated as a task-specific context at a different scale of features. Adapting the pretrained features with correlation-based residual attention enhances consistency and the discriminative nature of support and query features. This helps address the misalignment of classification weights and query vectors. To adapt the pretrained feature space, we use intermediate feature maps (layer 2 of Resnet12) f_i for a given sample in addition to the final layer feature vector (denoted as x_i) in episode T . Following the same ideology of Equation 1, we create a class representative feature map f_c^{avg} , using intermediate feature maps of the support samples for class c . where $f_c^{avg} \in \mathbb{R}^{M \times L \times P}$ and M, L, P represent the channel, height and width dimensions. To enhance the discriminative features based on the relationship between classes in a task, we introduce a class correlation attention module that generates a residual attention map based on the correlation map between a pair of intermediate feature vectors (flattened).

The correlation $corr_c^j$ between class c and $j, \forall j \in N \setminus c$, is computed as $corr_c^j = \tilde{f}_c^{avg^T} \cdot \tilde{f}_j^{avg}$. Here \tilde{f}_c^{avg} and \tilde{f}_j^{avg} are the L2-normalized versions of f_c^{avg}, f_j^{avg} respectively and have a dimension of $\mathbb{R}^{M \times L \times P}$. Since for every class c , the correlation is computed with other $N \setminus c$ classes, the class-specific correlation map for task T has a dimension of $(N - 1) \times LP \times LP$ (concatenation along the channel dimension), which is given by:

$$corr_c = [corr_c^j], \forall j \in Y_n \setminus c \quad (7)$$

where $[\cdot]$ is a concatenation operator.

We incorporate a convolutional attention module generator $CCAM(\cdot)$, that produces a residual attention map based on a correlation map input. This effectively accentuates unique characteristics in intermediate feature maps of class or query samples. The correlation of any class can be computed using Equation 7, which is passed to $CCAM(\cdot)$ module that generates the residual attention map A_c . It can be represented as follows:

$$A_c = CCAM(corr_c | \theta_{ccam}) \quad (8)$$

$$f'_c = f_c^{avg} \odot (1 + A_c) \quad (9)$$

where f'_c is the attentive feature map for class $c \in N$ for task T , f_c^{avg} represents the class representative feature map and θ_{ccam} refers to the parameters of $CCAM$ module. To learn a feature embedding that can grasp these attentive features, we employ a network that consists of 2 residual blocks and denote it as the Feature Refinement Module (FRM). FRM expects an input of dimension $(M \times L \times P)$ and outputs an embedding vector of dimension d , the same as that of the pretrained feature space:

$$w_c^{tsnr} = FRM(f'_c | \theta_{frm}) \quad (10)$$

where w_c^{tsnr} and f'_c represent the task-specific novel representation and the attentive feature map for class c , respectively. θ_{frm} are the learnable parameters of the $FRM(\cdot)$ module. We propose to use the following combination of embeddings to model our final classification vector:

$$w_c^{final} = (\lambda_{avg} \times w_c^{avg} + \lambda_{prior} \times w_c^{prior} + \lambda_{tc} \times w_c^{tc} + \lambda_{tsnr} \times w_c^{tsnr}) \odot g(s_c) \quad (11)$$

where $g(\cdot)$ represents the semantic attention generator we have adopted from [39]. Our final embedding space is a combination of the pretrained feature space and the task-specific representation space that we learn through attentive features. The classifier vectors are further enhanced with visual attention from semantic embeddings (GloVe embeddings of class names in this case).

3.3 Task-Specific Refined Query Embedding

To adapt the query features along with their corresponding classification weight vectors in the refined embedding space, we follow a similar process of refinement for the query samples as well. For every query sample x_i^q , we also retrieve its intermediate feature maps $f_i^q \in \mathbb{R}^{M \times L \times P}$. We follow the same ideology of class feature correlation and compute each query sample's correlation with N available classes in the task:

$$corr_i^j = \tilde{f}_i^{qT} \cdot \tilde{f}_j^{avg}, \forall j \in Y_n, \forall f_i^q \in D^q \quad (12)$$

$$corr_i = [corr_i^j], \forall j \in Y_n, \forall f_i^q \in D^q \quad (13)$$

Attentive Query Feature map refinement: Unlike the class-specific correlation map, the query-class correlation map has a dimension of $N \times LP \times LP$ since every query sample is compared with all the classes in the task. For query attentive features, we use a similar network such as $CCAM(\cdot)$ without sharing any parameters. We represent this query attention generator as $QCCAM(\cdot)$ that expects an input of dimension $N \times LP \times LP$ and generates an attention map of dimension $L \times P$, which is applied over the query feature map $f_i^q \in \mathbb{R}^{M \times L \times P}$. The process of query refinement is as follows:

$$A_i^q = QCCAM(corr_i | \theta_{qccam}) \quad (14)$$

$$f_i'^q = f_i^q \odot (1 + A_i^q) \quad (15)$$

$$x_i'^q = FRM(f_i'^q | \theta_{frm}) \quad (16)$$

where θ_{qccam} , θ_{frm} represent the trainable parameters of the query attention generator module $QCCAM(\cdot)$ and the proposed feature refinement module ($FRM(\cdot)$). A_i^q denotes the residual attention that is applied over the intermediate feature map f_i^q to retrieve attentive feature maps $f_i'^q$. $x_i'^q$ represents the attentive embedding in d -dimensional feature space. We model the final refined query embedding as follows:

$$x_i^{tsnr-q} = x_i^q + \lambda_q \times x_i'^q \quad (17)$$

where λ_q is a learnable scalar coefficient for the refined embedding. Due to lack of space, we have included the architecture of (Q)CCAM in the Supplementary (Figure 1).

The loss function used to train is a combination of normalized temperature scaled cross-entropy and orthogonal loss between the classifier weights:

$$L_{total} = -\frac{1}{|D_q|} \sum_{i=1}^{|D_q|} \log \frac{\exp^{\tau \cdot \cos(x_i^{tsnr-q}, w_{c_i}^{final})}}{\sum_{c=1}^{|Y_n|} \exp^{\tau \cdot \cos(x_i^{tsnr-q}, w_c^{final})}} + \lambda * \left(\sum_{j \in Y_n} (1 - w_j^{final} w_j^{final^T}) + \sum_{j \neq i} w_j^{final} w_i^{final^T} \right) \quad (18)$$

3.4 Training Strategy

We follow the existing training strategy in two stages. For successful transfer of prior information, the first stage of training consists of training the *Feature Extractor* on the base classes D_{base} , which follows the standard training paradigm for classification. After the first stage of training, the *Feature Extractor* is frozen and only used for extracting features (final and intermediate layers). The second stage of training is carried out in an episodic paradigm of training such as [39]. We train our proposed task-specific classifier weight generator, task-specific query attention generator, feature refinement module, and semantic attention generator (from [39]) during the second stage of training, in an episodic manner similar to [6, 39]. Both classification weight vectors and query features are refined with our *Feature Refinement Module* and classified using a *cosine classifier*. We also add an orthogonal regularization to the classification vectors to increase inter-class margins. Further details are provided in the Supplementary.

4 Experiments and Results

In this section, we evaluate our proposed method on four benchmark datasets and analyze the effectiveness of it.

4.1 Datasets

We use four benchmark datasets for evaluation - miniImagenet [32], tieredImageNet [24], CIFAR-FS [2] and CUB [33]. Except for CUB, we use Glove embeddings of label names as the semantic information. For CUB, we use the visual attributes provided by [33]. Further information on dataset splits, and image resolutions are shared in the Supplementary material (Table 1).

4.2 Implementation Details

All experiments are conducted in PyTorch framework. The backbone of our method, the *Feature Extractor*, is a ResNet-12 [8] architecture. The correlation-based attention generator (*CCAM/QCCAM*) has been implemented as a three-layer convolution network with Batch Normalization and ReLU applied after the first convolutional layer, pooling layer after the second convolutional layer and softmax operation after the third and the final convolutional layer. The *Feature Refinement Module* has been implemented as two Residual Blocks with filters {320, 640}. Further details on training hyperparameters and architecture are provided in the Supplementary.

4.3 Comparison with benchmarks

Since we follow the inductive learning framework, we compare our method’s performance against several popular and state-of-the-art inductive FSL frameworks. We evaluate for 5-way 1-shot and 5-way 5-shot settings on all four benchmark datasets. Our proposed method achieves the best performance for the 5-way 1-shot setting for all benchmark datasets (Table 1), demonstrating the significance of task-specific adaptation of feature space. For the 5-way 5-shot setting, our method has surpassed other baselines for CUB and CIFAR-FS datasets, showcasing its ability for fine-grained classification. One of the primary reasons for such behavior is the availability of shareable features across different classes.

Table 1. Performance comparison for 5W1S and 5W5S experiments. Our proposed method (refined space) consistently outperforms other SoTAs in the 5W1S settings. * methods’ results have been reproduced with our dataset settings. A-FSL (Pretrained space) and (Refined space) are defined in section 4.4. The best is in **bold**.

Methods	Backbone	Task Specific	miniImagenet		tieredImagenet		CUB		CIFAR-FS	
			5W1S	5W5S	5W1S	5W5S	5W1S	5W5S	5W1S	5W5S
ProtoNet [28]	Conv-4	No	49.42±0.78	68.20±0.66	53.31±0.89	72.69±0.74	64.42±0.48	81.82±0.35	55.5±0.7	72.0±0.6
LEO [25]	WRN-28-10	Yes	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09	-	-	-	-
SNAIL [17]	Resnet-12	Yes	55.71±0.99	68.88±0.92	-	-	-	-	-	-
TADAM [19]	Resnet-12	Yes	58.50±0.30	76.70±0.30	-	-	-	-	-	-
AM3 [37]	Resnet-12	No	65.30±0.49	78.10±0.36	69.08±0.47	82.58±0.31	-	-	-	-
AWGIM [7]	WRN-28-10	Yes	63.12±0.08	78.40±0.11	67.69±0.11	82.82±0.13	-	-	-	-
TriNet [3]	Resnet-18	Yes	58.12±1.37	76.92±0.69	-	-	69.61±0.46	84.10±0.35	-	-
MetaOptNet [15]	Resnet-12	No	62.64±0.61	78.63±0.46	65.99±0.72	81.56±0.53	-	-	72.0±0.7	84.2±0.5
Dynamic-FSL [6]	Resnet-12	No	62.81±0.27	78.97±0.18	68.55±0.31	83.95±0.21	-	-	-	-
SEGA [39]	Resnet-12	No	69.04±0.26	79.03±0.18	72.18±0.30	84.28±0.21	84.57±0.22	90.85±0.16	78.45±0.24	86.00±0.20
DeepEMD* [41]	Resnet-12	No	65.91±0.82	79.28±0.20	69.84±0.32	84.06±0.23	70.71±0.30	86.13±0.19	-	-
DeepBDC* [36]	Resnet-12	No	60.76±0.28	78.25±0.20	63.03±0.31	81.57±0.22	65.45±0.29	85.01±0.19	-	-
Distill [31]	Resnet-12	No	64.82±0.60	82.14±0.43	71.52±0.69	86.03±0.49	-	-	73.9±0.8	86.9±0.5
TPMN [35]	Resnet-12	Yes	67.64±0.63	83.44±0.43	72.24±0.70	86.55±0.63	-	-	75.5±0.9	87.2±0.6
DMF [38]	Resnet-12	Yes	67.12±0.46	81.54±0.31	71.89±0.52	85.96±0.35	-	-	-	-
LPE-FSL [40]	Resnet-12	No	68.28±0.43	78.88±0.33	72.03±0.49	83.76±0.37	85.04±0.34	89.24±0.26	74.88±0.45	85.30±0.35
FGFL [4]	Resnet-12	Yes	69.14±0.80	86.01±0.62	73.21±0.88	87.21±0.61	80.77±0.90	92.01±0.71	-	-
A-FSL (Pretrained space)	Resnet-12	Yes	69.19±0.23	79.89±0.19	72.17±0.29	83.98±0.21	86.29±0.21	91.23±0.16	78.21±0.25	87.20±0.20
A-FSL (Refined space)	Resnet-12	Yes	70.47±0.23	82.57±0.17	73.84±0.29	84.50±0.21	87.22±0.20	92.95±0.15	78.82±0.25	88.98±0.20

Table 2. Performance evaluation for a varying amount of task-specific information transfer in 5W1S setting. m denotes the out-of-class samples that some class c can learn information from. Most of the datasets in this table reflect significant changes in performance with different values of m . The best is in **bold**.

Datasets	Out-of-class Samples (m)		
	$m = 1$	$m = 2$	$m = 3$
miniImagenet	68.97 +/- 0.23	68.77 +/- 0.24	70.47 +/- 0.23
CUB	85.38 +/- 0.22	87.22 +/- 0.20	86.17 +/- 0.22
CIFAR-FS	77.89 +/- 0.25	78.13 +/- 0.25	78.82 +/- 0.25
tieredImagenet	72.76 +/- 0.29	73.14 +/- 0.29	73.84 +/- 0.29

Table 3. Ablation study of the effect of intermediate features on the refinement of feature space. Due to the presence of greater spatial details in Layer 2 features, all datasets consistently perform better with Layer 2 features.

Datasets	Intermediate Feature Layer of ResNet-12	
	Layer 2	Layer 3
miniImagenet	70.47 +/- 0.23	68.58 +/- 0.24
CUB	87.22 +/- 0.20	86.02 +/- 0.20
CIFAR-FS	78.82 +/- 0.29	78.16 +/- 0.25
tieredImagenet	73.84 +/- 0.29	73.39 +/- 0.29

4.4 Ablation Study

In this section, we analyze the effectiveness of every component of our proposed method.

Refinement of Pretrained Feature Space In section 1, we claim that the effectiveness of FSL methods is limited due to the fixed pretrained feature space. To validate our claim, we conduct experiments for both 5W1S and 5W5S settings for all datasets. To compare the results obtained when classification is performed on the pretrained feature space versus the refined feature space, we use the following two protocols to train and evaluate:

- Classification on pretrained feature space: The classification scores for class c are computed using w_c^{pt} as the classifier weights and x_i^q as the query feature vectors - $score_c(x_i^q) = \tau \cdot \cos(x_i^q, w_c^{pt} \odot g(s_c))$. In Table 1, "A-FSL (Pretrained space)" represents the results of this experiment.
- Classification on refined feature space: The classification scores for class c are computed using w_c^{final} as the classifier weights and x_i^{tsnr-q} as the query feature vectors - $score_c(x_i^q) = \tau \cdot \cos(x_i^{tsnr-q}, w_c^{final})$. In Table 1, "A-FSL (Refined space)" represents the results of this experiment.

From Table 1, we can observe that for the 5W1S setting, refinement of feature space improves the performance consistently which supports our claim. Refinement of feature space improves the performance of 5W1S setting by 1.28%, 0.93%, 0.61%, and 1.67% for miniImagenet, CUB, CIFAR-FS and tieredImagenet, respectively. Results for 5W5S improve by 2.68%, 0.52%, 1.72% and 1.78% for miniImagenet, tieredImagenet, CUB, and CIFAR-FS, respectively.

Learning from Out-of-Class Samples To evaluate the effect of transferring information from out-of-class instances, we conduct experiments with varying values of out-of-class samples (m) that share information and update prototypes for some class c . For the 5W1S setting, we evaluate performance against $m =$

$\{1, 2, 3\}$ (Table 2). As expected, there are significant changes in performance when the value of m changes for any given dataset. For CUB, $m = 2$ proves to be an optimal choice for improved performance. For all other datasets, $m = 3$ is optimal for the best performance. Determining the best value for m in a dataset appears to be a heuristic search, as even fine-grained datasets like CUB and CIFAR-FS do not exhibit a consistent pattern. This variation may be due to differences in image resolution affecting the pretrained feature representation.

Effect of Intermediate Feature Maps In section 3.2 and 3.3, we propose a correlation-based attention generator that assists in activating spatial features of the target object in an image when pretrained features lack certain discriminative features. To analyze the role played by different levels of intermediate features, we conduct experiments with intermediate layers from Residual Blocks 2 and 3 from Resnet-12 [8] and compare the performance for the 5W1S setting for all datasets. Results from Table 3 indicate that Layer 2 features are more successful at improving the feature space for classification. Due to the lack of space, we included the feature map visualization in the Supplementary section. Figure 3, 4, and 5 of Supplementary provide an explainable visualization of the process of attentive feature refinement of Layer 2 features for a test episode of the CUB dataset. Layer 2 features are more informative than Layer 3 due to their larger resolution. The presence of additional information helps to create localized attention over the intermediate features and refine the feature space to a generalizable space.

4.5 Visualization

We provide MDS [14] visualizations of the feature space learned by our baseline and proposed method. Figure 2 (a) represents the query, support, and classifier weights when only pretrained features are used to train and evaluate. Figure 2 (b) represents the feature embeddings in the refined feature space, which is obtained using Equations 11 and 17 for classification. Quantitatively, we have higher accuracy for feature space in Figure 2(b) as compared to the pretrained feature space in Figure 2(a). Visually, classifier weights are pulled toward the query samples' clusters for their corresponding classes and contribute to further alignment. Figures 3, 4, and 5 of Supplementary reflect how the attention generated by our proposed modules $CCAM(\cdot)$ and $QCCAM(\cdot)$ can highlight the class distinctive features with low activation in the original feature map. For intermediate features lacking in details (ex, "Ruby Throated Hummingbird" support sample in Figure 3, Supplementary), our attention module can recover class-specific details due to its residual nature. A similar pattern is also observed for the query sample of the same class.

5 Complexity Analysis

The primary cost of our method is in the proposed $QCCAM(\cdot)/CCAM(\cdot)$ module (please refer to the Supplementary for details). If the dimension of given

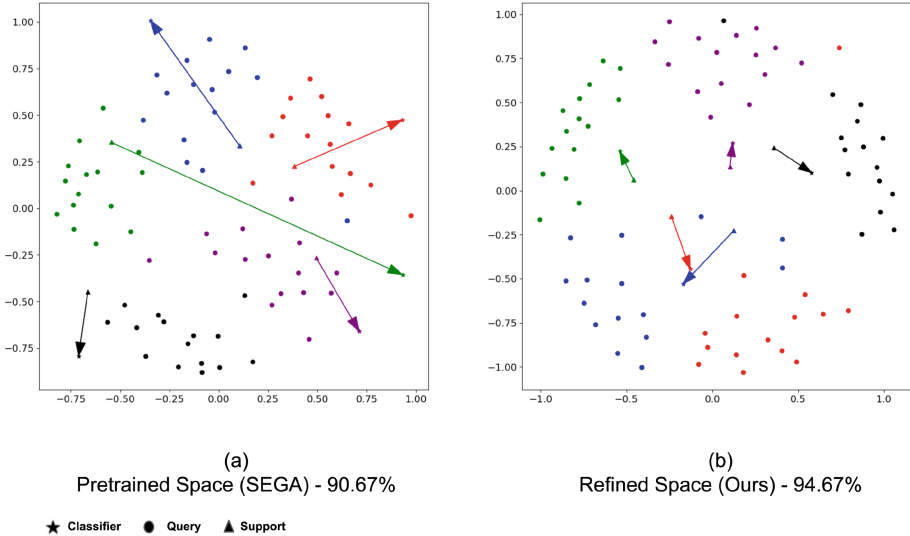


Fig. 2. Classification feature space comparison for CUB 5W1S. (a) Pretrained feature space from SEGA [39] for testing episode 3250 (b) Refined feature space from our proposed method for the same testing episode. The arrows are directed from the support sample to the classifier vectors.

feature maps is $M \times L \times P$, the time complexity is $\mathcal{O}(L^2 P^2 M)$, and the space complexity is $\mathcal{O}(LPM)$. In our method, we use Layer 2 intermediate features for all the datasets ($160 \times 21 \times 21$ for miniImagenet, CUB, tieredImagenet and $160 \times 16 \times 16$ for CIFAR-FS). Due to increased computation, second-stage training of the proposed method costs ~ 2.2 hours (0.3 second/training episode) on an NVIDIA Tesla V100 GPU for a miniImagenet 5-way 1-shot experiment.

6 Limitations

To address the high computation cost, we compute the correlation of query feature maps with averaged feature maps of support samples, which limits the effectiveness of our method to some extent. For the 5W1S setting, the complexity does not change. However, for 5W5S (or any multi-shot setting), the complexity increases exponentially if the correlation between query and individual support samples is computed. We plan to address this limitation in our future work by creating part-based localization using stronger task-specific/semantic-specific representations than correlation maps. Using part-based localization will assist in reducing the complexity as well as focusing on class-specific features when guided by semantic information. In our future work, we plan to adapt large vision and language models such as CLIP and GPT to improve feature representation and incorporate further generalization.

7 Conclusion

Our work aims to improve the performance of few-shot learning by creating meaningful task context representations and aggregating task-specific information. We claim and provide evidence that the refinement of pretrained feature space with attentive class-specific features is beneficial for aligning label clusters. Experimental results of the proposed spatial feature correlation-based attention generator demonstrate its ability to highlight class-specific details from task-specific spatial correlation and improve the pretrained feature space. We have also observed that this spatial feature correlation-based refinement can be transferred and generalized well to novel classes. In our future work, we plan to investigate the impact of powerful task contextual representations on few-shot learning problems, combined with the generalization power of recent vision language models such as CLIP and DINO.

Acknowledgement. This work was supported in part by the following grants: National Institutes of Health Grant RF1AG073424, National Institutes of Health Grant P30AG072980, and Arizona Department of Health Services Grant CTR057001. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

References

1. Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* **29** (2016)
2. Bertinetto, L., Henriques, J.F., Torr, P.H., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. *arXiv preprint [arXiv:1805.08136](https://arxiv.org/abs/1805.08136)* (2018)
3. Chen, Z., Fu, Y., Zhang, Y., Jiang, Y.G., Xue, X., Sigal, L.: Multi-level semantic feature augmentation for one-shot learning. *IEEE Trans. Image Process.* **28**(9), 4594–4605 (2019)
4. Cheng, H., Yang, S., Zhou, J.T., Guo, L., Wen, B.: Frequency guidance matters in few-shot learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 11814–11824 (2023)
5. Fei, N., Gao, Y., Lu, Z., Xiang, T.: Z-score normalization, hubness, and few-shot learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 142–151 (2021)
6. Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4367–4375 (2018)
7. Guo, Y., Cheung, N.M.: Attentive weights generation for few shot learning via information maximization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 13499–13508 (2020)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *CoRR* **abs/1512.03385** (2015), <http://arxiv.org/abs/1512.03385>
9. Hou, R., Chang, H., Ma, B., Shan, S., Chen, X.: Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems* **32** (2019)

10. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 831–839 (2019)
11. Kan, B., Wang, T., Lu, W., Zhen, X., Guan, W., Zheng, F.: Knowledge-aware prompt tuning for generalizable vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15670–15680 (2023)
12. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19113–19122 (2023)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
14. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika* **29**(1), 1–27 (1964)
15. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10657–10665 (2019)
16. Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., Hu, H.: Negative Margin Matters: Understanding Margin in Few-Shot Classification. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12349, pp. 438–455. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58548-8_26
17. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A simple neural attentive meta-learner. arXiv preprint [arXiv:1707.03141](https://arxiv.org/abs/1707.03141) (2017)
18. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems* **30** (2017)
19. Oreshkin, B., Rodríguez López, P., Lacoste, A.: Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems* **31** (2018)
20. Peng, Z., Li, Z., Zhang, J., Li, Y., Qi, G.J., Tang, J.: Few-shot image recognition with knowledge transfer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 441–449 (2019)
21. Qi, H., Brown, M., Lowe, D.G.: Low-shot learning with imprinted weights. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5822–5830 (2018)
22. Qiao, S., Liu, C., Shen, W., Yuille, A.L.: Few-shot image recognition by predicting parameters from activations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7229–7238 (2018)
23. Raghu, A., Raghu, M., Bengio, S., Vinyals, O.: Rapid learning or feature reuse? towards understanding the effectiveness of maml. arXiv preprint [arXiv:1909.09157](https://arxiv.org/abs/1909.09157) (2019)
24. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint [arXiv:1803.00676](https://arxiv.org/abs/1803.00676) (2018)
25. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-learning with latent embedding optimization. arXiv preprint [arXiv:1807.05960](https://arxiv.org/abs/1807.05960) (2018)
26. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International conference on machine learning. pp. 1842–1850. PMLR (2016)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)

28. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. *Advances in neural information processing systems* **30** (2017)
29. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1199–1208 (2018)
30. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1–9 (2015)
31. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking Few-Shot Image Classification: A Good Embedding is All You Need? In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12359, pp. 266–282. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_16
32. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. *Advances in neural information processing systems* **29** (2016)
33. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
34. Wang, Y.-X., Hebert, M.: Learning to Learn: Model Regression Networks for Easy Small Sample Learning. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9910, pp. 616–634. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_37
35. Wu, J., Zhang, T., Zhang, Y., Wu, F.: Task-aware part mining network for few-shot learning. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 8433–8442 (2021)
36. Xie, J., Long, F., Lv, J., Wang, Q., Li, P.: Joint distribution matters: Deep brownian distance covariance for few-shot classification. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 7972–7981 (2022)
37. Xing, C., Rostamzadeh, N., Oreshkin, B., O Pinheiro, P.O.: Adaptive cross-modal few-shot learning. *Advances in Neural Information Processing Systems* **32** (2019)
38. Xu, C., Fu, Y., Liu, C., Wang, C., Li, J., Huang, F., Zhang, L., Xue, X.: Learning dynamic alignment via meta-filter for few-shot learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 5182–5191 (2021)
39. Yang, F., Wang, R., Chen, X.: Sega: semantic guided attention on visual prototype for few-shot learning. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 1056–1066 (2022)
40. Yang, F., Wang, R., Chen, X.: Semantic guided latent parts embedding for few-shot learning. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. pp. 5447–5457 (January 2023)
41. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12203–12213 (2020)



Joint Modal Heterogeneous Balance Hashing for Unsupervised Cross-Modal Retrieval

Jie Zhang  and Mingyong Li 

School of Computer and Information Science, Chongqing Normal University,
Chongqing 401331, China
limingyong@cqnu.edu.cn

Abstract. Existing cross-modal hashing methods have made progress in enhancing retrieval capabilities and reducing model size, but they struggle to balance retrieval performance across different channels, leading to increased robustness. These methods often show low integration of multi-channel semantic information and fail to address image-text heterogeneity balance, focusing solely on enhancing retrieval accuracy, which can lead to high model robustness issues. We propose the Joint Modal Heterogeneous Balance Hashing for Unsupervised Cross-Modal Retrieval (JMBH) to address this. We utilise the large model CLIP to process raw data, facilitating multi-channel semantic integration. We then design multi-channel fusion modalities to explore co-occurrence information across channels and develop intra- and inter-channel constraints to mine this information. Extensive experiments on three datasets validate JMBH's effectiveness in balancing image-text heterogeneity and reducing robustness.

Keywords: Cross-Modal Hashing · Multi-Channel Semantic Integration · Balancing Image-Text Heterogeneity.

1 Introduction

The accelerated evolution of internet paradigms has increased the speed of information iteration, resulting in a massive influx of data. Concurrently, the forms of data have become diverse, including video, audio, text, images, and links. Consequently, the challenge of processing this vast amount of data stably and efficiently has garnered widespread research interest. In this context, low-cost storage and high computational efficiency have emerged as the preferred solutions in the market.

Hash encoding [28][9][16] has emerged as an effective method for addressing this issue. It maps instances of different modalities to Hamming space and utilises Hamming distance for computation, thereby significantly enhancing query speed and reducing storage requirements. Specifically, hash encoding methods compress high-dimensional data into low-dimensional binary codes, allowing for

quick matching of similar data during retrieval and substantial storage savings. This technique is particularly suitable for handling large-scale, multimodal datasets[14], enabling efficient cross-modal retrieval while maintaining accuracy.

As data volumes grow and multimodal data[15] becomes more diverse, the advantages of hash encoding in processing and retrieving big data are increasingly apparent. Hash encoding enhances retrieval speed and reduces storage costs, making it a preferred tool for researchers in the era of big data. Researching and optimising hash encoding to better handle complex, diverse data and large volumes is a key direction in information processing. Continuous innovation and improvement of hash algorithms can further enhance data processing and retrieval efficiency and accuracy, providing higher-quality technical support for various applications[6].

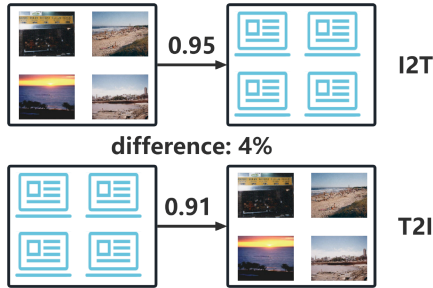


Fig. 1. The precision of image retrieval based on text is significantly higher than that of text retrieval based on images, leading to heterogeneity between images and text.

In retrieval, hash encoding is widely used due to its efficiency and low storage requirements. However, most researchers focus on improving retrieval performance, aiming for accuracy and lightweight solutions, often neglecting the balance of retrieval capabilities. For instance, a search engine that can retrieve text through images but not images through text is of limited practicality. Therefore, ensuring generality while maintaining high accuracy also holds significant research value. As shown in Figure 1, image retrieval has high precision in text, but text retrieval in images has low precision. To this end, our research focuses on balancing retrieval capabilities across different modalities while ensuring retrieval performance, ensuring that the model possesses robust cross-modal retrieval capabilities and reducing system robustness issues. Specifically, we explore how to effectively integrate and utilize modality-specific and modality-shared information in cross-modal retrieval to achieve broader application scenarios. For instance, by enhancing the features of image and text data, the system can better capture co-occurrence information. We design modality processing methods to balance cross-modal retrieval capabilities. This not only improves the practicality and flexibility of the system but also accommodates more diverse real-world application needs. Through this research, we aim to provide a more

comprehensive and balanced solution in the field of retrieval, bringing breakthroughs to cross-modal retrieval. In summary, the main contributions of this paper are as follows:

1. Utilizing pre-trained large models to process raw data features, obtaining high-quality image and text features.
2. Designing numerous image-text fusion modalities to uncover co-occurrence information across different channels. Developing intra-channel and inter-channel constraints to mine co-occurrence information.
3. We conducted comprehensive experiments on three widely used image and text retrieval datasets, validating that JMBH significantly balances modal heterogeneity while enhancing retrieval performance.

2 Related work

2.1 Deep cross-modal hashing

Supervised hashing methods partition data using labels to enhance retrieval performance. For example, Relaxed Energy Preserving Hashing for Image Retrieval (REPH)[14] proposes an energy preservation strategy that retains the original data’s energy in the transformed space, thereby mitigating the energy loss during hash projection. Self-Supervised Multi-Modal Knowledge Graph Contrastive Hashing for Cross-Modal Search (CMGCH)[3] constructs a multi-modal knowledge graph, representing implicit multi-modal knowledge relationships between images and text as inter-modal and intra-modal semantic associations. Weakly-Supervised Enhanced Semantic-Aware Hashing for Cross-Modal Retrieval (WASH)[3] jointly decomposes low-rank semantic factors and multi-modal features into a common subspace to reduce the heterogeneity gap, thereby enhancing the semantic awareness of shared representations.

Compared to supervised methods, unsupervised methods do not require labels to obtain information. They partition similarity based on the intrinsic semantic relationships within the data, offering significant advantages in terms of cost and application scenarios. Unsupervised Dual Hashing Coding on Semantic Tagging and Sample Content for Cross-modal Retrieval (UDC)[1] jointly learns dual hash codes for semantic tagging and sample content by decomposing the feature matching potential. By preserving consistent semantic information and cross-modal correlations, it bridges both the semantic and heterogeneous gaps between different modalities. Dual Self-Paced Cross-Modal Hashing (DSCMH)[13] mimics human cognitive learning by learning hash codes from “easy” to “difficult” at the instance and feature levels, thereby mitigating the adverse effects of noise or outliers. Scalable Unsupervised Hashing via Exploiting Robust Cross-modal Consistency (SUH)[6] discretely learns hash codes by exploiting robust consistency from latent semantic information and feature embeddings to avoid cumulative quantization loss. Multi-Relational Deep

Hashing for Cross-Modal Search (MRDH)[4] integrates comprehensive modelling of similarity relationships between different modal data to effectively bridge modality gaps.

2.2 Retrieval Based on Pre-trained Models

CLIP developed by OpenAI, is a large pre-trained model designed for contrastive learning using a vast amount of image-text pairs. It employs a unified model to handle both images and text, enabling understanding and generation of multimodal content related to vision and language. Trained on a dataset containing 400 million image-text pairs, CLIP[20][22][7] demonstrates strong zero-shot learning capabilities, allowing direct application to tasks such as image classification, image search, and text generation without specific fine-tuning. This positions CLIP as a significant innovation in the field of multimodal learning, greatly enhancing the fusion and understanding of visual and language signals.

3 Proposed method

We propose an unsupervised method that balances the heterogeneity between images and text, controlling the disparity in cross-modal retrieval capabilities within a minimal range without relying on labels to enhance retrieval performance. An overview is illustrated in Figure 2.

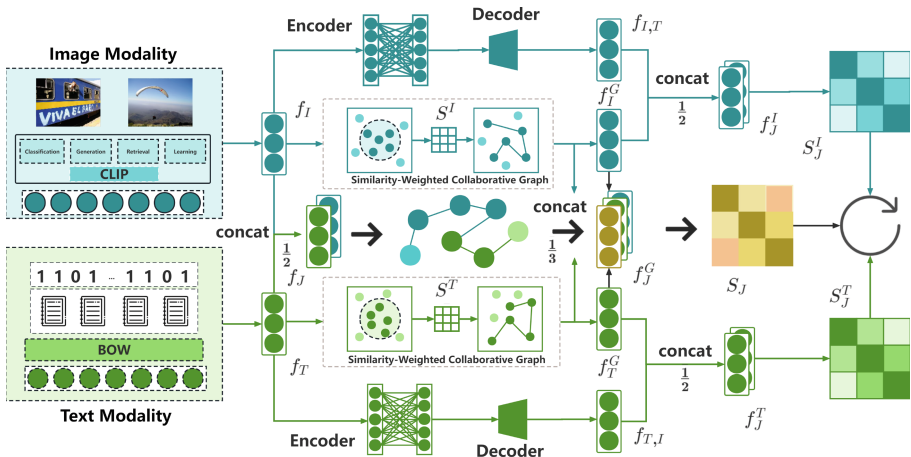


Fig. 2. Model Overview Diagram for Joint Modal Heterogeneous Balance Hashing for Unsupervised Cross-Modal Retrieval (JMBH).

3.1 Deep Feature Enhancement

Linear similarity matrix generation is achieved by integrating image and text information to form a matrix containing semantic information from both modalities. Specifically, we utilize the pre-trained model CLIP[29] to extract image features at multiple levels, including image classification, image-to-text generation, image retrieval, and zero-shot learning. This approach enables comprehensive semantic exploration from various perspectives and contextual links, maximising the model’s capability to enhance understanding and correlations between images and text. We use f_I to denote the image features extracted by CLIP.

In common unsupervised methods, the similarity matrix for text modality is typically computed using cosine similarity. However, this approach often neglects a significant amount of textual semantic information, thereby weakening the model’s performance in image retrieval based on text. Therefore, we utilize the CLIP model to convert words from the vocabulary into word embeddings. We obtained matrix $C = \{\{c_i\}_{i=1}^v\} \in R^{d \times v}$, d represents the dimensionality of each word embedding. We take the cosine similarity values between c_i and c_j greater than zero as our associated similarity r_{ij} .

3.2 Joint Modality Collaborative Construction

Compared to supervised methods, unsupervised methods establish correlations through features without requiring label information, which enhances adaptability in retrieval but also introduces noise. Therefore, in this section, we aim to maximize the exploration of shared semantics in data by leveraging a large amount of joint information from two modalities, enhancing the association of relevant instances and weakening that of irrelevant instances. Specifically, we first construct a semantic matrix using features extracted from both modalities, including new features from text. The expression is as follows:

$$S_1 = \lambda_1 S_{II} + (1 - \lambda_1) S_{TT} \quad (1)$$

S_{II} and S_{TT} are the semantic matrices generated from image and text features, respectively.

Next, we also prepare for fusion by generating semantic matrices corresponding to the reconstructed features, maximizing the exploration of semantic information in the features that couldn’t be uncovered through a single modality alone. Additionally, from another perspective — specifically, a non-linear perspective — we enhance the relationships between pairs of similarities. Through extensive experimentation, this plays a crucial role in balancing the retrieval capabilities of both modalities. The expression is as follows:

$$S_2 = \frac{2}{1 + e^{-S_1}} - 1 + I \quad (2)$$

$\mathbf{1}$ represents an all-one matrix, and I denotes the identity matrix. Through element-wise activation functions, we constrain inputs around 0 and 1, which

positively influences the determination of instance pairs. Finally, we integrate the two matrices organically. The expression is as follows:

$$S = \lambda_2 S_1 + (1 - \lambda_2) S_2 \quad (3)$$

3.3 Similarity Weighted Collaborative Graph

Based on our experiments, we found cases where scattered instance pairs exhibit fuzzy or even incorrect judgments. Therefore, we decided to preprocess the semantic matrices of the two modalities, correcting instance pairs with ambiguous or erroneous judgments to further enhance the relevance of the semantic matrix.

Influenced by JDSH[5], we discovered that using similarity values from the semantic matrices of both modalities provides a clearer distinction for edge cases. Thus, we adopt this approach to process the matrices from this perspective. Specifically, we first follow the idea of JDSH, distinguishing the two threshold endpoints by using the mean and standard deviation of Gaussian and Laplace distributions. Represented as μ_L, σ_L, μ_R and σ_R . Simultaneously, we set two thresholds $s_L = \mu_L - \eta_L \sigma_L$ and $s_R = \mu_R + \eta_L \sigma_R$. Thus, we compare the similarity value of each instance pair with the set thresholds. In summary, when the similarity value of an instance pair is greater than s_R , we determine them to be similar. When the similarity value is less than s_L , we determine them to be dissimilar.

When determined to be similar, we use W_+ to reduce the distance. When determined to be dissimilar, we use W_- to increase the distance. The expressions are as follows:

$$\begin{aligned} W_+ &= 1 + \alpha_1 e^{s_{ij} - s_{\max}} \\ W_- &= 1 + \alpha_2 e^{s_{\min} - s_{ij}} \end{aligned} \quad (4)$$

s_{ij} is the similarity value of the instance pair, s_{\max} is the value set for reducing the distance of similar instance pairs, and s_{\min} is the value set for increasing the distance of dissimilar instance pairs.

Thus, we obtained high-quality semantic matrices for both modalities. However, experiments have shown that the model’s performance is suboptimal at certain hash code lengths. Our analysis revealed that this is because the semantic matrices of a single modality contain only semantic information from a single similarity perspective. Consequently, some data do not perform well at specific hash code lengths. Therefore, we propose processing the matrices further by feeding them into a graph neural network for additional refinement.

Specifically, we leverage GCN to capture high-order semantic relationships between instances, maximizing the exploration of semantic information overlooked by linear structures from a graph perspective, and further enhancing both modalities. Specifically, we encode the aforementioned features and input their vector representations into the GCN. Each layer’s convolution process is illustrated as follows:

$$H_{(l)}^k = \sigma_{(l)}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H_{(l-1)}^k) W_{(l)}^k \quad (5)$$

Where W_l^k is the l th layer of the k -modality convolutional filter, σ is the activation function, and H represents the corresponding output.

After processing both modalities and extensively combining them, we have improved retrieval performance and achieved significant effectiveness in balancing mutual retrieval between modalities.

3.4 Hash Learning

After the features of two modalities undergo similarity-weighted collaborative graph construction, they are jointly processed. This approach enables the exploration of latent higher-order semantics between instances and promotes a balanced retrieval capability between images and text, stabilizing mutual retrieval within a narrow range.

Then, by jointly combining two vectors to generate the corresponding semantic matrix and , compared to the initial semantic matrix, it includes more diverse interactive semantics from different perspectives. Where we constrain the joint encoding reconstruction loss to achieve this. The expression is as follows:

$$L_{JER} = \|S_J - H(f_J^I, f_J^I)\|_F^2 + \|S_J - H(f_J^T, f_J^T)\|_F^2 + \|S_J - H(f_J^I, f_J^T)\|_F^2 \quad (6)$$

$H(*, *)$ denotes cosine similarity calculation between vectors.

In designing constraints, we aim to leverage the co-occurrence information implied by both modalities. We connect the features generated by the autoencoder and the similarity-weighted collaborative graph into a new feature. This preserves strong consistency within channels while simultaneously incorporating original semantics from another perspective after feature extraction. Therefore, we introduce the Joint Preservation Reconstruction Loss to constrain this:

$$L_{JPR} = \|\Gamma(H(f_J^I, f_J^T)) - 1.5E\|_F^2 \quad (7)$$

Γ denotes the matrix formed by the diagonal elements of a matrix, and E represents the identity matrix.

By extensively integrating features from image and text modalities and exploring their latent semantics, we handle single-modality processing and cross-channel interactions. This minimizes modal heterogeneity. Our experiments show this approach balances inter-modal heterogeneity and reduces robustness. Additionally, we explore semantics from the untreated semantic matrix and combine the two pieces of information. We designed the Original Channel Encoding Loss to constrain this:

$$L_{OCE} = \|S - H(f_I, f_I)\|_F^2 + \|S - H(f_T, f_T)\|_F^2 + \|S - H(f_J, f_J)\|_F^2 \quad (8)$$

We aggregate all losses into a unified learning framework, significantly enhancing efficiency. Therefore, our final objective function is as follows:

$$L = L_{JER} + \gamma_2 L_{JPR} + L_{OCE} \quad (9)$$

γ_2 is the balancing factor used to balance the various components of the loss.

4 Experiments

4.1 Evaluation datasets

MIRFlickr-25K: The dataset includes 25,000 pairs of image instances, of which we select 20,015 pairs with over 20 tags. For text, we use 1386-dimensional Bag-of-Words (BOW) vectors. From this dataset, 2000 instances are used for testing, while the remaining serve as the retrieval set. Additionally, 5000 instances are sampled from the retrieval set for training.

NUS-WIDE: The dataset comprises 269,648 images across 81 classes. We focus on 186,577 image instances from 10 classes, with 2000 pairs designated as the query set and the rest as the retrieval set. We select 5000 instances from the retrieval set for training. Text features are represented using 1000-dimensional Bag-of-Words (BOW).

MS COCO: The dataset includes 123,287 pairs of image instances, partitioned similarly to the other datasets. Text features are represented using a 2000-dimensional Bag-of-Words (BOW).

4.2 Evaluation criterion

We use Mean Average Precision (MAP) for analysis and comparison in our experiments. For MAP, if an image and text data point share one or more common labels, we consider them similar; otherwise, they are considered dissimilar.

4.3 Results Analysis

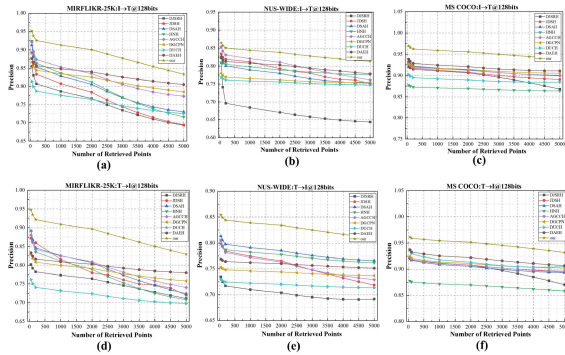


Fig. 3. (a), (b), and (c) represent the top-N precision curves for image retrieval with text queries ($I \rightarrow T$) on the MIRFLICKR-25K, NUS-WIDE, and MS COCO datasets, respectively. Similarly, (d), (e), and (f) correspond to the top-N precision curves for text retrieval with image queries ($T \rightarrow I$) on these three datasets.

Table 1. The mAP@50 performances of MIRFlickr-25K and NUS-WIDE datasets at various hashing code lengths.

Task	Method	MIRFLickr-25K			NUS-WIDE		
		16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
I2T	DJSRH[12]	0.812	0.841	0.865	0.721	0.776	0.794
	HNH[26]	0.856	0.885	0.895	0.586	0.802	0.818
	JDSH[5]	0.823	0.852	0.888	0.738	0.786	0.831
	DSAH[21]	0.869	0.878	0.892	0.773	0.802	0.816
	DGCPN[23]	0.851	0.869	0.884	0.785	0.815	0.821
	AGCH[25]	0.865	0.883	0.897	0.807	0.832	0.833
	DUCH[8]	0.848	0.866	0.873	0.754	0.778	0.814
	DAEH[10]	0.812	0.838	0.847	0.763	0.791	0.802
	JMBH	0.917	0.919	0.933	0.835	0.833	0.836
	T2I	DJSRH[12]	0.785	0.827	0.837	0.714	0.752
HNH[26]		0.833	0.856	0.865	0.436	0.779	0.795
JDSH[5]		0.828	0.869	0.876	0.724	0.784	0.793
DSAH[21]		0.841	0.857	0.884	0.775	0.796	0.807
DGCPN[23]		0.827	0.855	0.877	0.747	0.767	0.782
AGCH[25]		0.826	0.844	0.854	0.763	0.788	0.793
DUCH[8]		0.829	0.851	0.867	0.725	0.755	0.776
DAEH[10]		0.771	0.815	0.825	0.713	0.757	0.763
JMBH		0.908	0.912	0.928	0.828	0.831	0.834

Map Results Analysis: We conducted extensive experiments on three datasets and plotted the top-N accuracy curves, as shown in Figure 3. Table 1 presents the results of our method compared to other methods for the top 50 samples at different code lengths on two public datasets. To further verify the effectiveness of our method, we also conducted experiments with 5000 samples at different code lengths across various datasets, as shown in Table 2. By examining the data in these tables, we can easily observe that:

1. On three commonly used unsupervised datasets, even though we have done extensive work to balance image-text heterogeneity, we still achieve significant performance improvements over advanced unsupervised methods across different datasets. This improvement is particularly evident with lower sample sizes and shorter code lengths. This is primarily because the performance of large models in processing data decreases as the number of samples increases. Consequently, as the sample size increases, the quality of the features we obtain gradually decreases, leading to a corresponding decline in retrieval capability. Therefore, with an increase in sample size and code length, the improvement on each dataset diminishes.
2. By observing the data in the tables, unlike previous unsupervised methods, our ability to retrieve images using text has significantly improved. In cases of low sample size and short hash code length, our retrieval accuracy has increased by

Table 2. The mAP@5000 performances of MIRFLickr-25K, MS COCO and NUS-WIDE datasets at various hashing code lengths.

Task	Method	MIRFLickr-25K			MS COCO			NUS-WIDE		
		16 bit	32 bit	64 bit	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
I2I	IMH[11]	0.681	0.659	0.643	0.737	0.687	0.681	0.607	0.623	0.619
	LSSH[27]	0.675	0.677	0.682	0.813	0.832	0.838	0.678	0.706	0.703
	STMH[17]	0.566	0.585	0.619	0.646	0.687	0.653	0.409	0.429	0.468
	CMFH[2]	0.686	0.692	0.701	0.725	0.757	0.777	0.635	0.664	0.699
	FSH[8]	0.659	0.678	0.684	0.748	0.772	0.794	0.578	0.596	0.631
	RFDH[18]	0.636	0.648	0.658	0.691	0.711	0.749	0.551	0.572	0.608
	DJSRH[12]	0.673	0.701	0.734	0.754	0.815	0.861	0.587	0.671	0.717
	MGAH[24]	0.631	0.649	0.658	0.783	0.807	0.814	0.601	0.677	0.715
	JIMRH[19]	0.611	0.622	0.633	0.661	0.706	0.732	0.493	0.516	0.551
	JDSH[5]	0.725	0.731	0.752	0.692	0.758	0.888	0.678	0.724	0.743
	DSAH[21]	0.639	0.766	0.779	0.851	0.881	0.901	0.724	0.753	0.772
	DGCPN[23]	0.759	0.781	0.779	0.883	0.902	0.902	0.715	0.745	0.756
	HNH[26]	0.733	0.745	0.738	0.832	0.855	0.868	0.684	0.721	0.741
	DUCH[8]	0.667	0.688	0.706	0.847	0.866	0.876	0.686	0.714	0.728
	DAEH[10]	0.782	0.794	0.801	0.894	0.902	0.905	0.732	0.754	0.772
	JMBH	0.799	0.813	0.817	0.912	0.922	0.925	0.789	0.795	0.803
T2I	IMH[11]	0.681	0.667	0.654	0.768	0.717	0.715	0.626	0.644	0.638
	LSSH[27]	0.648	0.653	0.662	0.708	0.745	0.779	0.567	0.587	0.624
	STMH[17]	0.643	0.674	0.691	0.686	0.769	0.811	0.581	0.611	0.645
	CMFH[2]	0.661	0.669	0.679	0.757	0.789	0.809	0.609	0.641	0.672
	FSH[8]	0.682	0.697	0.702	0.769	0.791	0.809	0.609	0.649	0.665
	RFDH[18]	0.625	0.646	0.654	0.701	0.717	0.741	0.551	0.568	0.592
	DJSRH[12]	0.675	0.691	0.698	0.759	0.832	0.862	0.601	0.656	0.707
	MGAH[24]	0.627	0.648	0.625	0.747	0.772	0.768	0.591	0.613	0.645
	JIMRH[19]	0.647	0.647	0.657	0.728	0.767	0.779	0.584	0.586	0.612
	JDSH[5]	0.699	0.719	0.724	0.758	0.829	0.895	0.674	0.715	0.711
	DSAH[21]	0.646	0.754	0.759	0.854	0.886	0.890	0.668	0.716	0.748
	DGCPN[23]	0.727	0.751	0.757	0.881	0.897	0.899	0.702	0.723	0.742
	HNH[26]	0.723	0.721	0.706	0.839	0.863	0.867	0.671	0.699	0.696
	DUCH[8]	0.652	0.668	0.681	0.861	0.885	0.898	0.662	0.694	0.709
	DAEH[10]	0.762	0.767	0.774	0.888	0.899	0.901	0.713	0.733	0.748
	JMBH	0.802	0.812	0.827	0.913	0.929	0.931	0.771	0.784	0.801

more than 5 percentage points. This is mainly because, in processing text features, we extensively integrate semantic information from different channels, thereby further enhancing the accuracy within a single channel.

3. Most unsupervised methods focus solely on improving retrieval performance within different channels, neglecting the balance between channels' heterogeneity, which further exacerbates the differences between channels. Even if a single channel has high performance, it cannot maintain high performance across all channels. Therefore, we have done extensive work to enhance consistency between channels. Specifically, we integrated a significant amount of cross-channel semantic co-occurrence information. By examining the three tables, it is evident that the performance difference between our image and text channels is minimal. Instead of pursuing extremely high performance in a single channel, we aimed for balanced development across all channels, reducing robustness. This balanced approach holds substantial value for practical applications.

4.4 Ablation Study

To test the performance of each component of our method, we designed three variants as follows:

JMBH-1 The first variant removes deep feature enhancement, meaning it does not use the CLIP large model for data processing; instead, it uses unprocessed features. Observing Table 3, we can see a significant decline in the model's performance. This is primarily due to the large model's powerful data processing capabilities. Without this capability, the quality of the obtained features significantly decreases.

JMBH-2 Removing the similarity-weighted collaborative graph, we directly generate semantic matrices from features of both channels for loss calculation. Observing Table 3, the retrieval performance of the model significantly decreases, and the disparity in performance between different channels gradually increases. This is because we enhanced from the perspective of instance similarity, correcting misjudgments of instance pairs.

JMBH-3 Removing the joint modality, that is, excluding features from both channels such as f_J and f_J^G , we observe from the table that there is a significant increase in retrieval performance disparity between different channels. This is because losing the co-occurrence information from different modalities results in a large disparity in retrieval capability, further exacerbating modality heterogeneity.

4.5 Parameter Sensitivity Analysis

In this experiment section, we provide a detailed analysis of two hyperparameters in our method, which significantly impact the performance of our model. We conducted experiments on the MIRFlickr-25K, NUS-WIDE, and MS COCO datasets, selecting the top 50 samples for the experiments. As shown in Figure 4.

4.6 Visualization

In Figure 6, We validated our method on the MS COCO dataset. For intuitive visualization, we annotated the detected objects in the images. The red

Table 3. Conducting ablation experiments on the three datasets with varying code lengths, using mAP@5000 as the evaluation metric.

Task	Method	MIRFLickr-25K			
		16 bit	32 bit	64 bit	128 bit
I2T	JMBH-10	0.768	0.789	0.804	0.814
	JMBH-20	0.784	0.798	0.805	0.789
	JMBH-30	0.775	0.795	0.807	0.816
	JMBH	0.799	0.813	0.817	0.833
T2I	JMBH-10	0.759	0.785	0.795	0.808
	JMBH-20	0.776	0.788	0.800	0.799
	JMBH-30	0.759	0.772	0.778	0.789
	JMBH	0.802	0.812	0.827	0.831

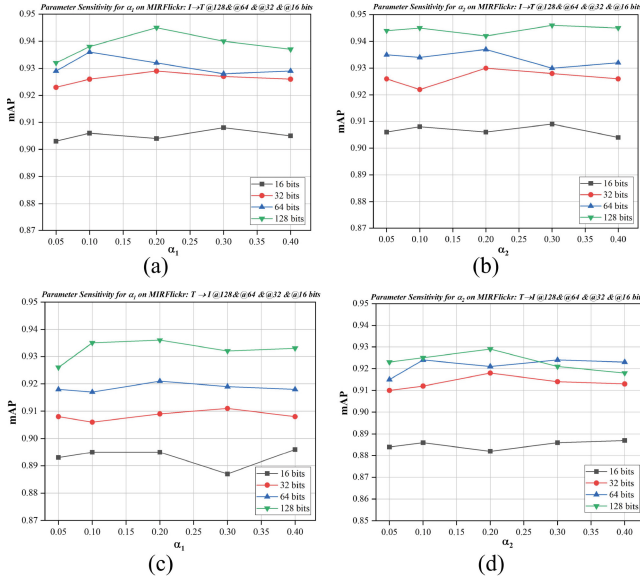


Fig. 4. Sensitivity analysis of α_1 and α_2 with different bit lengths for JMBH on MIRFlickr-25K.

text represents the keywords for text-to-image retrieval. Figure 5 shows the loss convergence of our model.

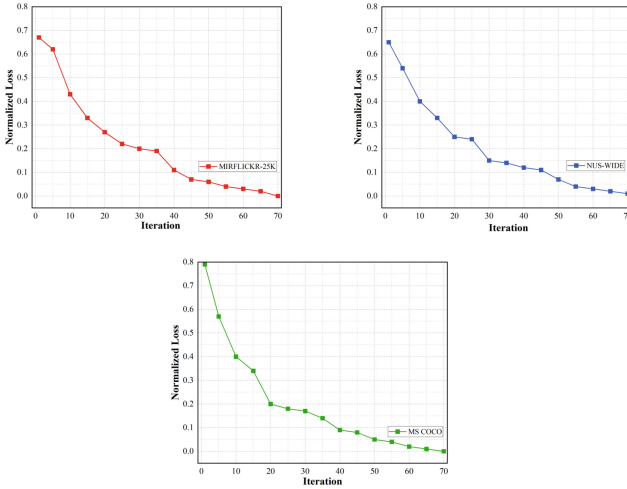


Fig. 5. Loss convergence curves for JMBH.

<p>A male football player wearing a blue jersey.</p>	
<p>An orange-red motorcycle is travelling on the racetrack.</p>	
	<p>A man holding a tennis racquet in his hand standing on a tennis court. A man flexing his arm and holding a tennis racket and tennis ball on a tennis court. A man is standing on a tennis court ready to play. A man on a tennis court who is swinging at a tennis ball. A man swings his racket on a tennis court.</p>
	<p>An elephant standing in the grass by itself. A large elephant is standing in a field. An elephant in a grass and tree area. An elephant walks in an open grassy field. An elephant taking cover from the sun under an area with a canopy.</p>

Fig. 6. Visualization of image-text retrieval results on MS COCO.

5 Conclusion

This paper introduces a novel Joint Modal Heterogeneous Balance Hashing for Unsupervised Cross-Modal Retrieval. Initially, the CLIP model processes raw data to extract high-quality feature representations from both channels. These encoded features are then decoded to enhance semantic information retrieval. Additionally, the Similarity-Weighted Collaborative Graph strengthens similarity matrices between image and text channels. Features from multiple channels are

integrated, and this fused information is used to construct constraints. Extensive experiments validate the efficacy of our approach in achieving balanced representation across image and text domains.

Acknowledgements. This work was supported by the Chongqing social science planning project(Grant No. 2023BS085).

References

1. Cai, H., Zhang, B., Li, J., Hu, B., Chen, J.: Unsupervised dual hashing coding (udc) on semantic tagging and sample content for cross-modal retrieval. *IEEE Transactions on Multimedia* (2024)
2. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2075–2082 (2014)
3. Liang, M., Du, J., Liang, Z., Xing, Y., Huang, W., Xue, Z.: Self-supervised multimodal knowledge graph contrastive hashing for cross-modal search. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 13744–13753 (2024)
4. Liang, X., Yang, E., Yang, Y., Deng, C.: Multi-relational deep hashing for cross-modal search. *IEEE Transactions on Image Processing* (2024)
5. Liu, S., Qian, S., Guan, Y., Zhan, J., Ying, L.: Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. pp. 1379–1388 (2020)
6. Liu, X., Li, J., Nie, X., Zhang, X., Wang, S., Yin, Y.: Scalable unsupervised hashing via exploiting robust cross-modal consistency. *IEEE Transactions on Big Data* (2024)
7. Ma, H., Zhao, H., Lin, Z., Kale, A., Wang, Z., Yu, T., Gu, J., Choudhary, S., Xie, X.: Ei-clip: Entity-aware interventional contrastive learning for e-commerce cross-modal retrieval. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 18051–18061 (2022)
8. Mikriukov, G., Ravanbakhsh, M., Demir, B.: Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing. *arXiv preprint arXiv:2201.08125* (2022)
9. Mingyong, L., Yewen, L., Mingyuan, G., Longfei, M.: Clip-based fusion-modal reconstructing hashing for large-scale unsupervised cross-modal retrieval. *International Journal of Multimedia Information Retrieval* **12**(1), 2 (2023)
10. Shi, Y., Zhao, Y., Liu, X., Zheng, F., Ou, W., You, X., Peng, Q.: Deep adaptively-enhanced hashing with discriminative similarity guidance for unsupervised cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **32**(10), 7255–7268 (2022)
11. Song, J., Yang, Y., Yang, Y., Huang, Z., Shen, H.T.: Inter-media hashing for large-scale retrieval from heterogeneous data sources. In: *Proceedings of the 2013 ACM SIGMOD international conference on management of data*. pp. 785–796 (2013)
12. Su, S., Zhong, Z., Zhang, C.: Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 3027–3035 (2019)
13. Sun, Y., Dai, J., Ren, Z., Chen, Y., Peng, D., Hu, P.: Dual self-paced cross-modal hashing. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 15184–15192 (2024)

14. Sun, Y., Dai, J., Ren, Z., Li, Q., Peng, D.: Relaxed energy preserving hashing for image retrieval. *IEEE Transactions on Intelligent Transportation Systems* (2024)
15. Sun, Y., Xue, H., Song, R., Liu, B., Yang, H., Fu, J.: Long-form video-language pre-training with multimodal temporal contrastive learning. *Adv. Neural. Inf. Process. Syst.* **35**, 38032–38045 (2022)
16. Tu, R.C., Jiang, J., Lin, Q., Cai, C., Tian, S., Wang, H., Liu, W.: Unsupervised cross-modal hashing with modality-interaction. *IEEE Transactions on Circuits and Systems for Video Technology* (2023)
17. Wang, D., Gao, X., Wang, X., He, L.: Semantic topic multimodal hashing for cross-media retrieval. In: *Twenty-fourth international joint conference on artificial intelligence* (2015)
18. Wang, D., Wang, Q., Gao, X.: Robust and flexible discrete hashing for cross-modal similarity search. *IEEE Trans. Circuits Syst. Video Technol.* **28**(10), 2703–2715 (2017)
19. Wang, D., Wang, Q., He, L., Gao, X., Tian, Y.: Joint and individual matrix factorization hashing for large-scale cross-modal retrieval. *Pattern Recogn.* **107**, 107479 (2020)
20. Xia, X., Dong, G., Li, F., Zhu, L., Ying, X.: When clip meets cross-modal hashing retrieval: A new strong baseline. *Information Fusion* **100**, 101968 (2023)
21. Yang, D., Wu, D., Zhang, W., Zhang, H., Li, B., Wang, W.: Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In: *Proceedings of the 2020 international conference on multimedia retrieval*. pp. 44–52 (2020)
22. Yu, H., Ding, S., Li, L., Wu, J.: Self-attentive clip hashing for unsupervised cross-modal retrieval. In: *Proceedings of the 4th ACM International Conference on Multimedia in Asia*. pp. 1–7 (2022)
23. Yu, J., Zhou, H., Zhan, Y., Tao, D.: Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 35, pp. 4626–4634 (2021)
24. Zhang, J., Peng, Y.: Multi-pathway generative adversarial hashing for unsupervised cross-modal retrieval. *IEEE Trans. Multimedia* **22**(1), 174–187 (2019)
25. Zhang, P.F., Li, Y., Huang, Z., Xu, X.S.: Aggregation-based graph convolutional hashing for unsupervised cross-modal retrieval. *IEEE Trans. Multimedia* **24**, 466–479 (2021)
26. Zhang, P.F., Luo, Y., Huang, Z., Xu, X.S., Song, J.: High-order nonlocal hashing for unsupervised cross-modal retrieval. *World Wide Web* **24**, 563–583 (2021)
27. Zhou, J., Ding, G., Guo, Y.: Latent semantic sparse hashing for cross-modal similarity search. In: *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. pp. 415–424 (2014)
28. Zhu, L., Wu, X., Li, J., Zhang, Z., Guan, W., Shen, H.T.: Work together: correlation-identity reconstruction hashing for unsupervised cross-modal retrieval. *IEEE Transactions on Knowledge and Data Engineering* (2022)
29. Zhuo, Y., Li, Y., Hsiao, J., Ho, C., Li, B.: Clip4hashing: Unsupervised deep hashing for cross-modal video-text retrieval. In: *Proceedings of the 2022 international conference on multimedia retrieval*. pp. 158–166 (2022)



Interpretable Visual Semantic Alignment via Spectral Attribution

Shivanvitha Ambati[✉], Vineet Padmanabhan[✉], Wilson Naik Bhukya,
and Rajendra Prasad Lal

School of Computer and Information Sciences, University of Hyderabad,
Hyderabad 500046, Telangana, India
shivanvitha21@gmail.com, {vineetnair,rathore,rajendraprasd}@uohyd.ac.in

Abstract. The initial Transformer architecture which was introduced for text, has been extended to image, speech and other domains. Multimodal models which combine more than one kind of data, and vision-language models in particular, have also seen increasing adoption. The interpretability of these models is crucial due to their potential for subtle errors and their diverse applications. Existing interpretability methods for Transformers primarily employ attention maps to explain vision-language alignment. This overlooks the contribution from other parts of the transformer block like Layer Normalization and Feed-Forward Network (FFN) and can lead to incorrect image and text segment attribution to the model's decision. We propose an approach that mitigates this issue by using the output of the transformer modules instead of attention maps as the basis for deriving the interpretability vectors. We use Spectral Graph Theory and propose three variants of our method, namely: DSMI (Deep Spectral Method for Interpretability), DSMI + Grad (DSMI with gradients) and DSMI + Grad + Attn (DSMI with gradients & attention maps). Each version has its own advantages with varying performance based on the class of models which are being analyzed. We show with detailed experiments that our methods are superior to some of the existing interpretability techniques such as GradCAM and have comparable interpretability to methods like LRP and other state-of-the-art methods while being simpler to implement.

Keywords: Transformer · Interpretability · Spectral · Bimodal

1 Introduction

The demand for interpretable AI¹ has roots that date back to early symbolic AI techniques like Bayesian networks and expert systems [23]. However, the emergence of deep learning has significantly propelled this field of research due to the

¹ We use interpretability and explainability interchangeably depending on the context

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_9.



Fig. 1. Relevance mask for “What is reflecting in the building’s windows?”

inherent complexity of these models, especially multimodal models. Transformers are increasingly being used to tackle multimodal tasks involving reasoning, such as Visual Question Answering [2] and Image-Text Matching. This is because transformers have the ability to realize the relationships between different types of information through mechanisms like self-attention and co-attention. These models attempt to align the vision and language components in their latent representations. However, these representations and their alignments are highly entangled and therefore, difficult to interpret. In critical tasks like medical analysis, high speed locomotion, financial data analysis etc., interpreting this alignment can be crucial for understanding and debugging the model results.

Current bimodal models may be classified according to the 1) type of image encoder and 2) variant of the co-attention mechanism used by these models for fusion. For example, models like LXMERT [29] and VisualBERT [16] rely on frozen object detectors like FasterRCNN [22] to retrieve image embeddings but contain different co-attention mechanisms, i.e., Bidirectional Cross-Attention and Merged Attention respectively, for aligning the vision and text embeddings. Others, like METER [9] and ViLT [12], use patch-based features extracted from ViTs [8] with Bidirectional Cross-Attention and Merged Attention, respectively. All of these models use some pre-trained text encoder to obtain the corresponding text embeddings e.g., BERT [7], RoBERTa [17], GPT [21] etc. Certain bimodal models like ALBEF [15] use more advanced approaches to align the modalities e.g., contrastive learning like in CLIP [20] and cross-attention based fusion to strengthen the relationship between image and text information.

Unimodal models have a more established body of work regarding interpretability and there are several methods which originated for explaining CNNs, e.g. CAM, GradCAM, LRP [3, 10, 25] that were ported directly for interpreting transformer-based unimodal and multimodal models. Transformer specific methods like Rollout [1] attempt to analyze the Attention mechanism and these are also applied to understanding multimodal models, with some adjustments to account for the multimodal attention mechanisms. Interpretability methods like GradCAM [25] and Rollout [1] often produce inaccurately localized explanations for multimodal models and other methods like LRP [3, 4] are computationally intensive and quite complex to implement. There has been recent work [5, 6] for better post hoc interpretability of Transformer-based models and multimodal

models in particular, by distinguishing the way self-attention layers and co-attention layers are handled.

We attempt to continue in that direction and focus specifically on generating visual explanations of both the image modality and the text modality in multimodal Transformer based models. We utilize concepts from Spectral Graph Theory [28] to delineate the salient features at specific blocks of the model. The models on which we experiment, are trained on the VQA [2] task, though it can be any vision-language task. Our method is model-agnostic to some extent and capable of generating post-hoc local explanations catering to each image-text pair provided for inference. The job of our interpretability module is to highlight the regions of image and text that contribute to the model’s decision.

We make the following contributions in this paper:

1. We propose three different variants of our method based on Spectral Graph Theory which can be adapted to specific multimodal Transformer architectures
2. Our method is unsupervised in nature and it can generate class-specific and class-agnostic explanations according to the variant used
3. We conduct detailed experiments on vision-language, bimodal Transformers and determine the suitability of each variant of our method for specific Transformer models
4. Finally we also provide all the code of experiments for reproducibility at [Transformer-spectral-interpretability](#) and [METER-spectral-interpretability](#)

The remainder of the paper is structured as follows. Section 2 explores some of the related work and their shortcomings along with an overview of how we arrived at our method. Section 3 discusses the core methodology, the algorithms for our approach, and their significance. In Section 4, we showcase the application of our method on two bimodal models with visual examples. In Section 5, we compare our approach with existing methods using a quantitative evaluation test for both the bimodal models considered for experimentation. We also discuss the limitations of our approach and see how our method adheres to basic interpretability properties. Finally, Section 6 concludes the paper and proposes potential future research directions.

2 Related Work

2.1 Interpretability Techniques

Most transformer based models are explained via raw attention maps. This approach is often inaccurate and fails to capture the contributions from other components of the transformer like the intermediate output comprising contributions from FFN and Layer Normalization. There are intrinsically explainable models [27] stemming from sparse reconstruction and a special kind of ViT called AbSViT. But intrinsically interpretable models are hard to train requiring a lot of time and resources. One of the most famous class-specific methods is Grad-CAM [25]. The drawback with this method is that it fails to localize accurately.

Designed for CNNs where features are weighted by gradients, it does not translate well to transformers. In transformers, instead of features, attention maps [30] are weighed with their gradients yet the method falls short. Another well-known explanation technique is LRP [3, 4]. This method is class-agnostic by default, but several class-specific versions were also introduced. The biggest drawback with this method is that though it is post-hoc in nature, it is 1.5x slower than other methods which makes it sensitive to batch size during inference. Similar to our method, Spectral Relevance Analysis (SpRAy) [13] leverages spectral theory on CNNs. However, SpRAy focuses on global explanations, analyzing heatmaps across the entire dataset to identify potentially misleading patterns. Although this provides valuable insight into biases in training data, it can be cumbersome for individual data points. Our approach prioritizes local explanations, offering a more efficient way to understand how the model interprets specific inputs. While the current best method by Hila Chefer et al. [5] achieves strong performance, it has limitations. This class-specific method excels by leveraging all attention heads and layers, but does not utilize the full transformer architecture, namely the last block’s Layer Normalization and FFN. Furthermore, its relevance rules, designed for each attention type, hinder seamless integration across different models. The dependence on the modality holding the Classification ([CLS]) token and the model’s architecture limits its flexibility and raises confusion.

2.2 Spectral Approach

Recently, one line of research has emerged focusing on methods from Spectral Graph Theory, which studies the spectral properties of the Adjacency Matrix or Laplacian Matrix corresponding to the graph. Since the weights and features at each layer are large matrices, their spectral analysis can reveal interesting insights into their inner workings. Prior work harnessing Spectral Theory [13] focused on global explanations of the entire dataset. Recently Melas-Kyriazi et al [19] proposed a straightforward method for image segmentation and object localization based on Spectral Graph Theory. Their method detects the salient image features within the outputs of a Transformer using eigenvectors of the Laplacian matrix. Melas-Kyriazi et al., demonstrates effective unsupervised object localization and segmentation which is at par with state-of-the-art methods while focusing on a specific type of Transformer, namely ViT [8]. We take inspiration from their approach and adapt it to explain the decisions of multimodal models in an unsupervised manner. Our work differs from theirs however, in that we propose new methods with the Spectral Graph Theory as a basis, and focus entirely on bimodal Transformers. The details of our method are given in the next section.

3 Method

Bimodal Transformers like LXMERT [29], METER [9], use cross-attention or fusion to align intermediate vision and language features. This alignment can naturally be thought of as a bipartite graph, although the alignment is latent in

the weight matrices and the features. We use the features of the cross-modality blocks present in bimodal models, and discover the most salient features and their alignment by leveraging the eigenvectors of the Laplacian Matrix. These fusion blocks usually contain cross-attention layers, self-attention layers and FFN. To put it in a formal manner, let the deep features of one modality, M from the last fusion block be:

$$x_m = \phi(M) \in R^{n \times d_h}$$

where ϕ is the network, x_m are the features of modality M , and n, d_h are the number of tokens in the modality and hidden size respectively. The affinity matrix of the features is then, W_{feat} , where,

$$W_{feat} = x \cdot x^T = \begin{cases} xx^T[i, j] & x[i, j] \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The Laplacian (L) of W_{feat} is calculated as:

$$L = D - W, \text{ OR } L = D^{-1/2}(D - W)D^{-1/2} \quad (2)$$

In Equation (2), the matrix D is the diagonal matrix where the diagonal elements are sum of rows in W_{feat} , i.e., $D \in R^{n \times n}$, $D_{ii} = \sum_j W_{ij}$, and $D_{ij} = 0, \forall i \neq j$. The segments of the image/text are the eigenvectors $\{y_0, \dots, y_{n-1}\} = eigs(L)$ of the Laplacian $L = D^{-1/2}(D - W)D^{-1/2}$ of the feature affinity matrix W_{feat} . The eigenvector associated with the second smallest eigenvalue, called the **Fiedler Eigenvector** [26], captures the graph’s strongest connections. In image analysis, this translates to pinpointing the most visually informative areas. In text analysis, it focuses on extracting the most significant words. Algorithm 1 computes the Fiedler Eigenvector for a given feature Laplacian matrix with $eigs$ being responsible for solving (3),

Algorithm 1. DSMI

```

procedure GET_RELEVANCY(feats)
   $W_{feat} \leftarrow feats \cdot feats^T$ 
   $W_{feat} [W_{feat} < 0] \leftarrow 0$ 
   $L \leftarrow D - W_{feat}$ 
  eigenvalues, eigenvectors  $\leftarrow eigs(L)$ 
  return eigenvectors[1]
end procedure

```

$$LX = \lambda DX \quad (3)$$

Pure graph spectral method is class-agnostic in nature. Applied to interpretability of bimodal models, this method solely examines the fusion module in bimodal models. It does not shed light on the overall decision-making process. This is where the role of gradients comes into picture to make DSMI class-specific. The gradients of the attention maps are calculated w.r.t. the model

output (an answer in case of VQA). We propose incorporating the gradients of self-attention weights into the spectral approach to make it more robust. For this purpose, we put forth a hybrid graph spectral interpretability technique in which we take the product of the gradients of self-attention maps and Fiedler Eigenvector for every block in the fusion module and sum them up as shown in Algorithm 2. Algorithm 1 and Algorithm 2 are not transformer-specific. To introduce transformer specificity, we propose Algorithm 3 where we weigh the self-attention maps of each head with their gradients [6] and average across all heads to obtain an aggregate attention map. We then multiply the aggregate map by the Fiedler Eigenvector obtained in Algorithm 1 and accumulate contributions from every block in the cross-modality encoder.

Algorithm 2. DSMI + Grad

```

procedure GET_RELEVANCY_DSM_GRAD(feats_list)
  n_tokens ← feats_list[0].size(0)
  grad_fev ← [0]n_tokens
  for i, feats ∈ enumerate(feat_list) do
    fev = GET_RELEVANCY(feats)
    grad ← GET_GRAD_ATTEN_PROBS(i)
    grad_fev ← grad_fev + grad · fev
  end for
  return grad_fev
end procedure

```

Algorithm 3. DSMI+Grad+Attn

```

procedure GET_RELEVANCY_DSM_GRAD_CAM(feats_list)
  n_tokens ← feats_list[0].size(0)    ▷ Get the number of tokens in a modality
  gradcam_fev ← [0]n_tokens           ▷ Initialise the resultant tensor
  for i, feats ∈ enumerate(feat_list) do
    fev = GET_RELEVANCY(feats)
    grad ← GET_GRAD_ATTEN_PROBS(i)
    cam ← GET_ATTEN_PROBS(i)
    cam ← grad ⊙ cam                  ▷ Calculate the hadamard product
    gradcam_fev ← gradcam_fev + cam.mean() · fev
  end for
  return gradcam_fev
end procedure

```

4 Experiments

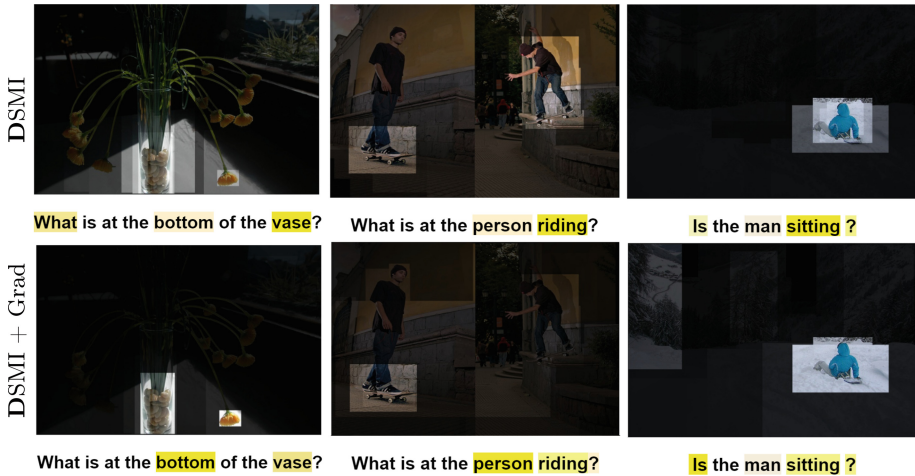


Fig. 2. Comparison between DSMI and DSMI + Grad for LXMERT: DSMI + Grad more accurately localizes the relevant regions in the images and the relevant words in the questions when compared to DSMI; Answers (left to right): “rocks”, “skateboard”, “yes”

For our experiments we focus on two primary models, each having a different kind of image encoder and text encoder. We also compared the visualizations of two of the spectral approaches for both models in Figures 2 and 3. The first model we examine is LXMERT [29]. It consists of a frozen object detection model, Faster RCNN [22] as the object relationship encoder and BERT [7] as the text encoder. It has self-attention modules for both modalities and a cross-modality encoder consisting of bidirectional cross-attention layers, self-attention layers and FFN. For our spectral approaches, we use features from the cross-modality encoder and gradients of self-attention weights of the same encoder. The second model we examine is METER [9]. Unlike LXMERT, it consists of a ViT-B/16 [8] as the image encoder and RoBERTa [17] as the text encoder. It consists of 6 blocks each containing a self-attention layer, followed by a bidirectional cross-attention layer and FFN. We consider the features of these blocks and the gradients of self-attention maps of the same blocks for our methodology.

The gradient based graph spectral method shines when it comes to accuracy of localization for both of the models. Especially if we consider Figure 3, in the second column, we see how DSMI + Grad was able to accurately deduce where the answer has been derived from in the image, i.e. the make of the laptop instead of highlighting the whole laptop like in DSMI. Further visualizations comparing our methods with the existing ones can be found in the Supplementary Material (Figures 2 and 3). Overall for METER, we see that class-specific DSMI + Grad

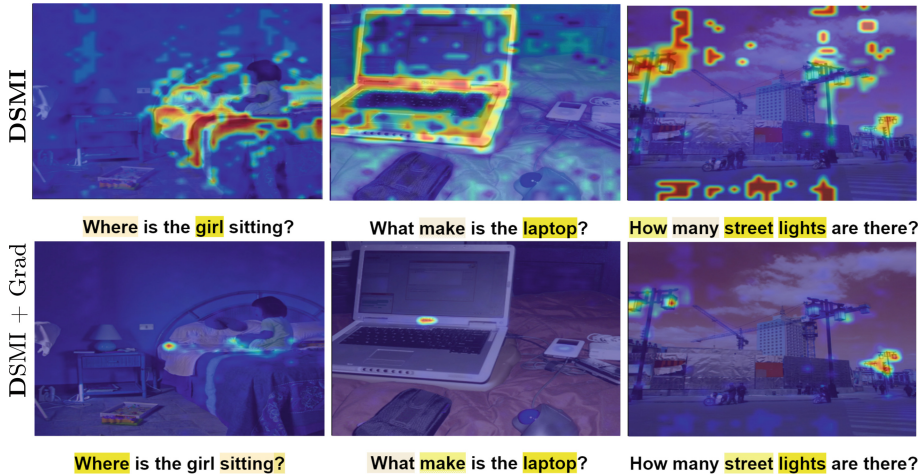


Fig. 3. Comparison between DSMI and DSMI + Grad for METER: DSMI + Grad is less noisy and provides more accurate heatmaps than DSMI; Answers (left to right): “bed”, “dell”, “5”

approach solved 3 key issues faced in DSMI: unclear heatmaps, noisy heatmaps, localization of small regions in the images. Another advantage that the DSMI + Grad approach gives us is that it can be applied to models like CLIP [20] to explain Image Text Matching, visualizations of which have been added to our Supplementary Material (Figure 1). Since CLIP does not have a cross-modality learning aspect, we observed that the standard DSMI approach would not produce optimal relevance scores. However, by incorporating gradients, we were able to extract richer information from the combined image and text data. While not achieving nearly optimal performance like in METER, our method demonstrates promising results in generating text-guided heatmaps for images in CLIP.

Similar to [5] we used VQA 2.0 dataset to evaluate our methods. It contains 265016 images from the MS COCO dataset, 5.4 open-ended questions on average per image and 10 answers per image-question pair. We performed perturbation tests to evaluate the interpretability methods by randomly sampling 3303 image-question pairs from the VQA 2.0 validation set. In positive perturbations, tokens are removed from highest to lowest relevance. A steep decrease in performance of the model is expected because important tokens were removed. In negative perturbations, tokens are removed from lowest to highest relevance and accuracy of the model drops very slowly because unimportant tokens are removed first.

5 Results and Discussion

DSMI + Grad + Attn and DSMI + Grad take the lead among the three variants of our method in LXMERT and METER respectively, as shown in Tables 1, 2.

Table 1. AUC for perturbations tests on LXMERT: Lower AUC for positive tests and higher AUC for negative tests indicates better performance Best - **Bold green**; Second best - **Bold black**; Third best - Underlined

Method	Image	Image	Text	Text
	+ve	-ve	+ve	-ve
Relevance Maps [5]	51.01	62.91	21.57	48.31
Transformer Attribution [6]	52.80	61.16	21.89	47.80
LRP [3,4]	<u>52.83</u>	60.82	24.14	44.51
Raw Attention	54.36	61.34	32.77	37.40
Grad-CAM [25]	57.97	59.59	34.50	37.56
Rollout [1]	57.15	58.26	38.71	31.47
DSMI	53.47	59.13	28.67	40.28
DSMI + Grad	53.77	62.28	24.78	44.80
DSMI + Grad + Attn	53.42	<u>62.23</u>	<u>24.12</u>	<u>45.47</u>

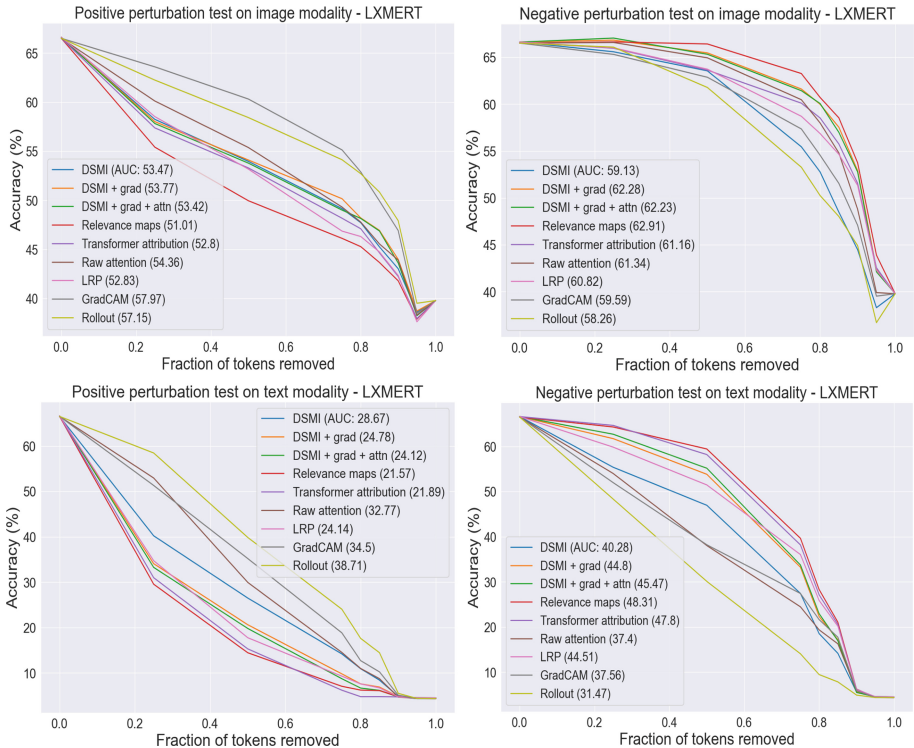


Fig. 4. Perturbation tests on LXMERT: Overall “Relevance maps” remains the leader followed by Transformer attr. The three DSMI variants have a performance comparable to LRP with DSMI + Grad surpassing Transformer attr. in -ve image perturbation test. DSMI + Grad + Attn surpasses LRP in 3/4 tests.

Table 2. AUC for perturbations tests on METER: Lower AUC for positive tests and higher AUC for negative tests indicates better performance Best - **Bold green**; Second best - **Bold black**; Third best - Underlined

Method	Image	Image	Text	Text
	+ve	-ve	+ve	-ve
Relevance Maps [5]	55.27	82.97	31.70	61.33
Transformer Attribution w/o LRP	<u>55.26</u>	82.96	31.72	61.29
Raw Attention	55.33	82.49	36.72	54.41
Grad-CAM [25]	70.95	73.52	49.21	49.83
Rollout [1]	59.96	82.17	44.32	49.93
DSMI	60.07	79.47	45.14	48.05
DSMI + Grad	54.45	<u>82.93</u>	<u>34.75</u>	<u>57.64</u>
DSMI + Grad + Attn	55.45	82.45	40.37	52.38

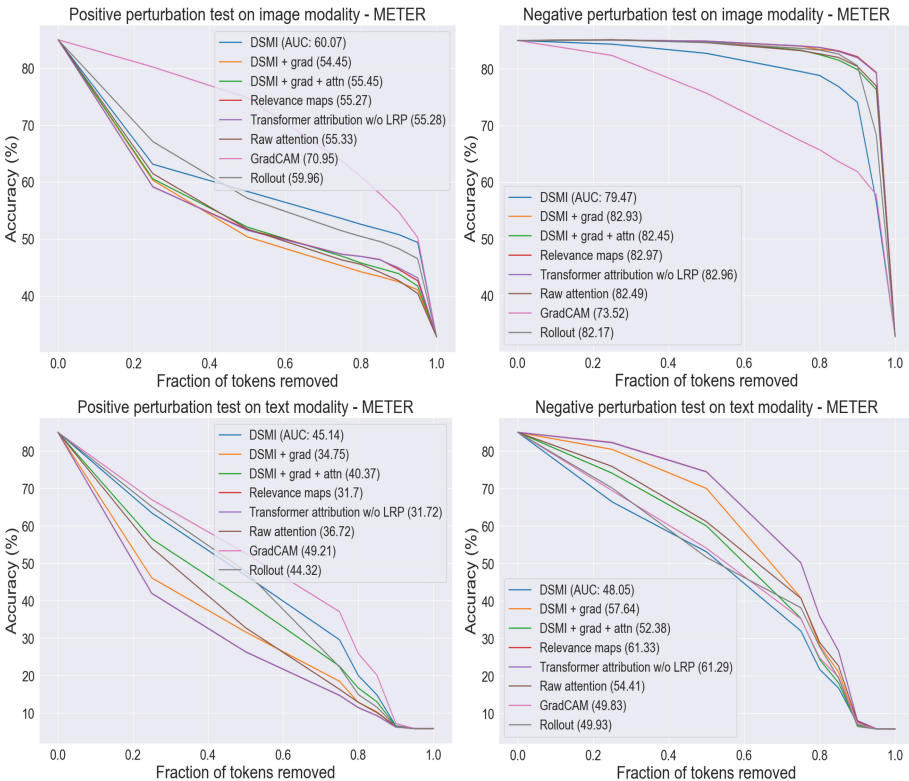


Fig. 5. Perturbation tests on METER: Overall “Relevance maps” remains the leader. The AUC for DSMI + Grad in +ve image perturbation test surpasses that of the state-of-the-art relevance maps approach. Transformer attr takes the second place with DSMI + Grad closely following.

5.1 Insights

The graph in Figure 4 shows that the accuracy of LXMERT significantly drops to 39.79% when image information is removed, but plummets to 4.52% without text tokens. This highlights the dependence of LXMERT on the unimodal contributions of the text, similar to the findings in [18]. Even with no image given, LXMERT retains some accuracy, suggesting the model leverages knowledge available in the dataset like humans might, based on the knowledge from the surroundings, to answer questions about unseen images. A similar pattern is also observed in METER (Figure 5). The textual bias is perfectly captured by both the bimodal models.

5.2 Limitations

Our approach excels when the model architecture uses bidirectional cross-attention. This allows us to generate explanations for both visual and language aspects. While CLIP lacks this module, we can still provide explanations using a gradient-based spectral approach because of the availability of gradients for both modalities. However, encoder-decoder architectures with unidirectional cross-attention, such as BLIP [14] with its ITM task, limit explanations to one modality (language in this case). The final image embeddings remain static after retrieval, preventing gradients for the vision component, and thereby preventing visual explanations. This is why we emphasize our method to be “fusion-specific” rather than “model-specific”.

5.3 Fulfillment of LRP Properties

LRP [4] talks about three properties that a good interpretability technique should satisfy. These properties themselves act as evaluation criteria to verify the validity of the explanations produced by a method. Therefore, we evaluated our methods against these properties to determine whether the methods exhibit a basic interpretability behavior.

1. **Conservation:** Any explanation generated by the model should be reflected somewhere in the input features
2. **Selectivity:** The model’s explanation should directly support its prediction. If evidence used for explanation is removed from the input, the model’s confidence in the prediction should decrease
3. **Continuity:** For similar input pairs that result in similar predictions, the explanations provided by the model should also be highly similar. This ensures that the explanation reflects the underlying reasoning process and is not random

The first property of conservation is already the basis of the three DSMI variants, as the underlying features of the models are used to calculate the relevance. The second property of selectivity holds for perturbation tests, where the model’s accuracy alters based on the kind of tokens that are removed. We

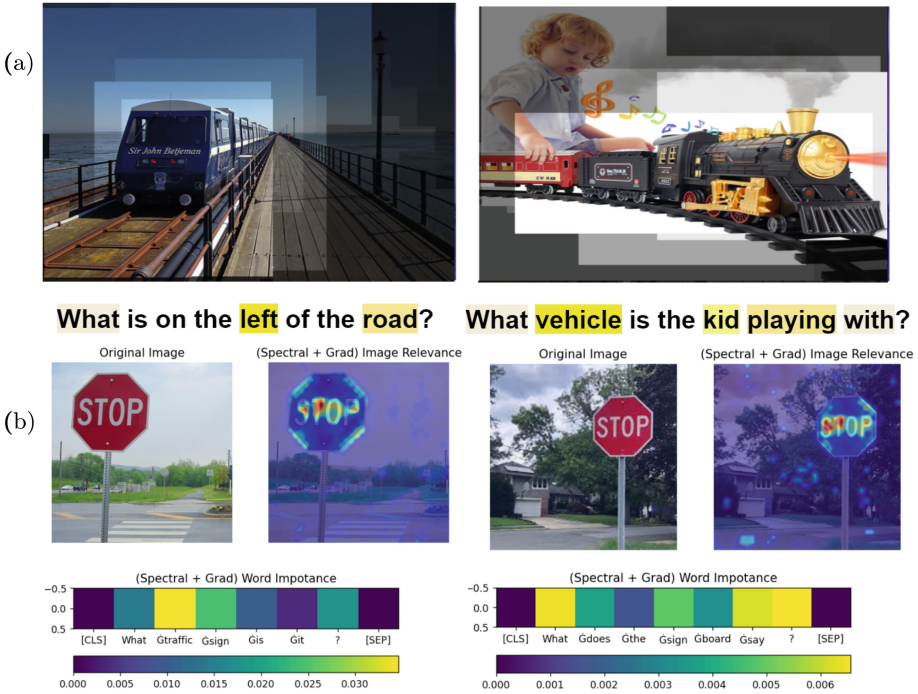


Fig. 6. Test for continuity: (a) LXMERT Answer: “train”; Our method accurately highlights the term “left” in the question which corresponds to the train in the image as the answer signifies (b) METER Answer: “stop”; DSMI + Grad is able to accurately focus on the text present on the traffic sign boards in the images.

investigated if our methods also satisfy the third property, continuity. For this we examined LXMERT and METER with similar (not identical) image-question pairs to assess the performance of one of the DSMI variants, DSMI + Grad across the models. Given a similar pair of an image and a question, our method is able to consistently pinpoint where the model derives its answer from the image, as shown in Figure 6. This suggests that our gradient-based graph spectral method exhibits continuity.

6 Conclusion and Future Work

The growing popularity of DNNs highlights the critical need for explainability techniques to ensure transparency. This paper proposes unique methods leveraging graph spectral theory, to explain bimodal models. We showcase how these methods, DSMI, DSMI + Grad and DSMI + Grad + Attn, offer post-hoc, localized visualizations of relevance specific to how the different modalities are fused. Although our approaches achieve slightly lower performance compared to the relevance maps introduced by Hila Chefer et al. [5], they offer distinct advantages.

Notably, DSMI and DSMI + Grad are highly adaptable to various models with minimal modifications as the methods depend only on output of an encoder’s blocks, i.e. features and gradients. These features have an accumulation of contributions from all components of the transformer blocks in a cross-modal encoder. Two of the DSMI variants are not transformer specific as we do not use attention maps, making them applicable to a wider range of architectures including non-transformer based architectures. Furthermore, DSMI and DSMI + Grad spectral methods demonstrate performance comparable to LRP while potentially mitigating LRP’s known issue of slower computation speed and DSMI + Grad + Attn outperforms LRP in case of LXMERT. In contrast to this, DSMI + Grad demonstrates performance very similar to the state-of-the-art relevance maps approach for image modality in METER as shown in Table 2.

We observe that our methods perform better on the image modality than the text modality. While this behavior can be overlooked for tasks like VQA where the answer depends on the image, it is crucial to give equal importance to both modalities in tasks like Image Text Matching. Future contributions can focus on handling text features differently. Although we were able to explain models with bidirectional cross-attention and contrastive learning, we were unable to replicate the results for models using merged attention such as ViLT [12] and VisualBERT [16]. Other interpretability techniques also seem to fall short on such models. Therefore this could be a possible future research direction. Another possible research contribution could be the birth of a new quantitative evaluation method for interpretability techniques. This is because perturbation tests are not sensitive to localization [24]. In the literature, the research for interpretability methods in case of bimodal models is heavily focusing on Image-to-Text models. Therefore, Text-to-Image models [11] should also be given more consideration for explainability while leveraging the unidirectional cross-attention process between prompt embeddings and the features of the intermediate denoised images.

References

1. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4190–4197. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.385>, <https://aclanthology.org/2020.acl-main.385>
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
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 (2015), <https://api.semanticscholar.org/CorpusID:9327892>
4. Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., Samek, W.: Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers. In: Villa, A.E.P., Masulli, P., Pons Rivero, A.J. (eds.) ICANN 2016. LNCS, vol. 9887, pp. 63–71. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44781-0_8

5. Chefer, H., Gur, S., Wolf, L.: Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 397–406 (October 2021)
6. Chefer, H., Gur, S., Wolf, L.: Transformer interpretability beyond attention visualization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 782–791 (2021)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1423>, <https://aclanthology.org/N19-1423>
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
9. Dou, Z.Y., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., Liu, Z., Zeng, M.: An empirical study of training end-to-end vision-and-language transformers. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022), <https://arxiv.org/abs/2111.02387>
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Adv. Neural. Inf. Process. Syst.* **33**, 6840–6851 (2020)
12. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 5583–5594. PMLR (18–24 Jul 2021), <http://proceedings.mlr.press/v139/kim21k.html>
13. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking clever hans predictors and assessing what machines really learn. *Nat. Commun.* **10**(1), 1096 (2019)
14. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022)
15. Li, J., Selvaraju, R.R., Gotmare, A.D., Joty, S., Xiong, C., Hoi, S.: Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS (2021)
16. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. arXiv preprint [arXiv:1908.03557](https://arxiv.org/abs/1908.03557) (2019)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
18. Lyu, Y., Liang, P.P., Deng, Z., Salakhutdinov, R., Morency, L.P.: Dime: Fine-grained interpretations of multimodal models via disentangled local explanations. In: Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society. p. 455–467. AIES '22, Association for Computing Machinery, New York, NY, USA (2022). <https://doi.org/10.1145/3514094.3534148>, <https://doi.org/10.1145/3514094.3534148>

19. Melas-Kyriazi, L., Rupprecht, C., Laina, I., Vedaldi, A.: Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8364–8375 (2022)
20. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
21. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI (2018), <https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf>
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
23. Saeed, W., Omlin, C.: Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Know.-Based Syst.* **263**(C) (mar 2023). <https://doi.org/10.1016/j.knosys.2023.110273>, <https://doi.org/10.1016/j.knosys.2023.110273>
24. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.): Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-28954-6>
25. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
26. Shao, H., Mesbahi, M.: On the fiedler vector of graphs and its application in consensus networks. *Proceedings of the American Control Conference* **2015**, 1734–1739 (07 2015). <https://doi.org/10.1109/ACC.2015.7170983>
27. Shi, B., Darrell, T., Wang, X.: Top-down visual attention from analysis by synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2102–2112 (2023)
28. Spielman, D.A.: Spectral graph theory and its applications. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science. p. 29–38. FOCS '07, IEEE Computer Society, USA (2007). <https://doi.org/10.1109/FOCS.2007.66>
29. Tan, H., Bansal, M.: LXMERT: Learning cross-modality encoder representations from transformers. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5100–5111. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1514>, <https://aclanthology.org/D19-1514>
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf



Progressive Learning Based on QP Distance for Enhancing HOP In-Loop Filter

Penghao Fu¹, Cheolkon Jung^{1(✉)}, Yang Liu², and Ming Li²

¹ School of Electronic Engineering, Xidian University, Xian 710071, China
zhengzk@xidian.edu.cn

² Guangdong OPPO Mobile Telecommunications Corporation, Dongguan, China

Abstract. To train the High-performance Operation Point (HOP) in-loop filter in VVC, the Joint Video Exploration Team (JVET) provides a three-stage training strategy that uses datasets compressed by the HOP embedded VVC Test Model (VTM) for training in the last two stages. However, the use of the HOP-embedded VTM to compress the training set twice more brings huge time-consumption. To address this issue, we propose progressive learning based on QP distance to enhance HOP in-loop filter while accelerating the HOP training time. We adopt the progressive learning strategy based on QP distance to strengthen the HOP learning ability. Based on QP distance, the proposed method does not use the training sets compressed by the HOP-embedded VTM, thus leading to remarkable reduction of training time. Moreover, the uncompressed video frames do not contain compression artifacts, thus the direct use of the uncompressed video data as label for training is not effective in capturing the relationship between the compressed input and its uncompressed label. However, based on QP distance, the proposed method uses higher-quality (lower QP setting) compressed data as label for training rather than using the uncompressed data as label, thus strengthening the HOP learning ability of removing compression artifacts. Experimental results show that the HOP model generated by the proposed method achieves an average BD-rate gain of -8.31% (Y), -16.28% (U), and -18.27% (V) over the VTM-11.0 anchor in the All Intra (AI) configuration thanks to the progressive learning based on the QP distance. Moreover, the proposed method reduces total training time to only 10 days while the three-training strategy recommended by JVET takes about 45 days.

Keywords: Progressive learning · compression artifact removal · high-performance operation point · in-loop filter · neural network · VVC · QP distance.

This work was supported by the National Natural Science Foundation of China (No. 62111540272).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15326, pp. 144–155, 2025.
https://doi.org/10.1007/978-3-031-78395-1_10

1 Introduction

The Versatile Video Coding (VVC) is the latest video coding standard investigated by the Joint Video Experts Team (JVET) from the ITU-T SG 16/Q.6 Video Coding Experts Group (VCEG) [12]. Its primary aim is to achieve exceptional compression efficiency and elevate video quality. The encoding process using the VVC Test Model (VTM) mainly includes intra prediction, inter prediction, transform and quantization, entropy coding, and in-loop filters. To enhance video quality, reduce compression artifacts, and optimize compression efficiency, VVC employs the in-loop filtering module that consists of five in-loop filters [4] as follows. The Deblocking Filter (DBF), which is dedicated to reducing block artifacts, serves as the primary filter. Sample Adaptive Offset (SAO) is the second filter designed to minimize ringing artifacts by finely adjusting the captured intensity changes. The Adaptive Loop Filter (ALF) is then introduced to rectify signal values based on the linearly filtered samples complemented by the Cross-Component Adaptive Loop Filter (CC-ALF). Finally, a specific filter [23], named Luma Mapping with Chroma Scaling (LMCS), is not explicitly intended for reducing blocking artifacts but rather focuses on exploiting the signal range to enhance coding efficiency [12]. This ensemble of in-loop filters within the VVC framework is elaborately designed to address various artifacts and optimize both video quality and coding efficiency by leveraging signal characteristics. The in-loop filters primarily rectify prediction residuals within the encoder by eliminating artifacts and pseudo-details caused by prediction, thereby enhancing video quality. The loop filters in VTM are manually designed based on signal processing theory and the assumption of stationary signals.

Recently, the neural network-based in-loop filters (NNLF), particularly those based on Convolutional Neural Networks (CNNs), exhibit stronger representation capabilities at the feature level in videos [10, 15, 22]. They excel in eliminating compression artifacts and outperform traditional in-loop filters in terms of performance. JVET recommends two NNLF architectures: High-performance Operation Point (HOP) [2] and Low-complexity Operation Point (LOP) [17]. Both architectures consist of three components: Head for shallow feature extraction, backbone for deep feature extraction, and tail for reconstruction. The network simplicity helps to reduce the number of model parameters and training time. The early form of NNLF architecture was proposed by Tencent in JVET-W0131 [18], and JVET-X0052 [19], which introduced depthwise separable convolution and standard convolution in the network. JVET-Z0091 [20] employed a single model in the filter design, which incorporated auxiliary information such as predicted frame, QP slice and QP base into the network. Several proposals, including JVET-AD0211 [13], integrated multi-scale feature extraction components into the backbone residual block. JVET-AD0106 [14], JVET-AD0166 [6], JVET-AD0168 [21], and JVET-AD0205 [7] contributed to the lightweight structure of the HOP model. JVET-AD0380 [2] proposed a general architecture for HOP in-loop filter that is comprised of shallow feature extraction, deep feature extraction, and reconstruction. HOP in-loop filter is concurrently trained by

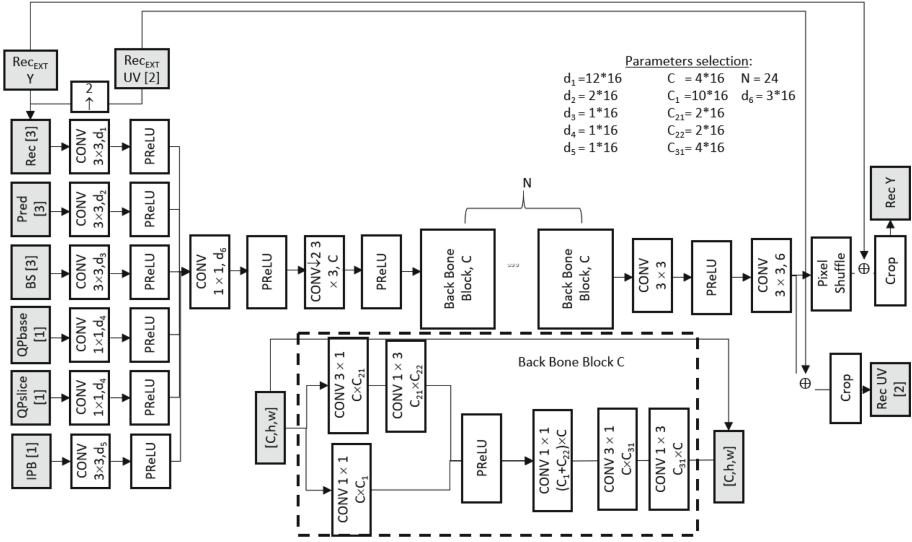


Fig. 1. Network architecture of the HOP in-loop filter released by JVET [2]. The HOP network architecture is composed of three parts: Head, backbone and tail. The head is used for shallow feature extraction and downsampling, the backbone is composed of several residual blocks for deep feature extraction, and the tail is used for upsampling and reconstruction.

multiple companies and cross-checked for its performance test. Fig. 1 shows the whole framework of HOP in-loop filter released by JVET [2].

Regarding the training process of HOP in-loop filter, JVET suggests using a three-stage training strategy [8,9] as shown in Fig. 2. In Stage I, the original VTM is used to compress the training set, while in Stage II and Stage III, the HOP-embedded VTM, i.e. 'VTM+CNN' in Fig. 2, is used to compress the training set. However, since the three-training strategy uses the HOP-embedded VTM to compress the training set twice more, it causes huge amount of training time. In this paper, we propose QP distance-leveraged acceleration of HOP training to save training time while improving the HOP model performance. We adopt a progressive learning strategy based on QP distance to strengthen the HOP learning ability. Without the use of the HOP-embedded VVC, we only use the original VTM once to generate the training set during the whole training process, thus remarkably saving training time. Moreover, we adopt progressive learning that removes compression artifacts step-by-step based on QP distance and significantly improves the model performance. Fig. 3 illustrates the proposed progressive learning strategy based on QP distance. In the first three steps (Model I to Model III), the QP distance increases by 5, while in the final step (Model IV), the labels are set to 7. Experimental results show that the proposed method achieves average BD rate gains of -8.31% (Y), -16.28% (U),

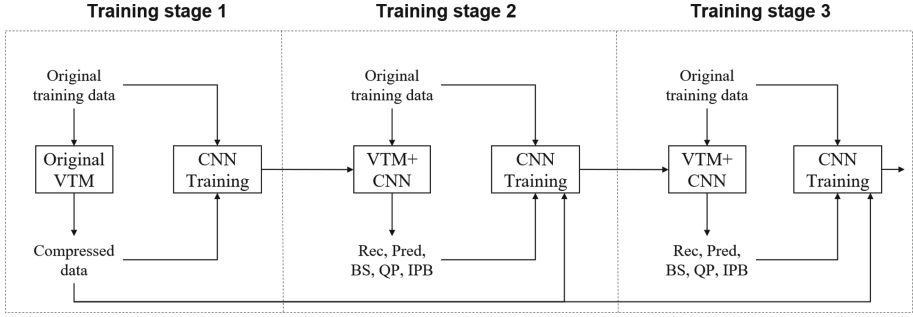


Fig. 2. Three-stage training strategy for HOP model generation recommended by JVET [8, 9]. VTM: VVC Test Model. CNN: Convolutional Neural Network. BS: Boundary structure. QP: Base QP and slice QP. IPB: Intra/inter prediction block.

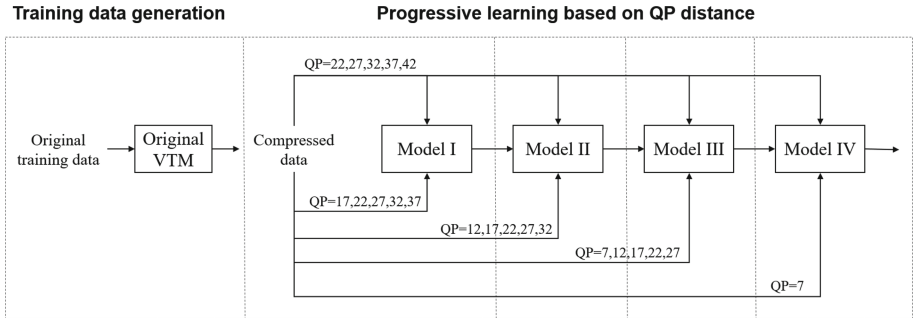


Fig. 3. Proposed progressive learning strategy based on QP distance. In the first three steps (Model I to Model III), the QP distance increases by 5, while in the final step (Model IV), the labels are set to 7.

and -18.27% (V) over the VTM-11.0 anchor in the All Intra (AI) configuration and the total training time is about ten days.

Compared with the three-stage training strategy recommended by JVET [8, 9], the main contributions of the proposed method are described as follows:

- During the whole training process, we only once generate the training set using the original VTM. However, the three-stage training strategy needs to generate the training set twice more with the HOP-embedded VTM. Therefore, the proposed method significantly reduces the training time.
- Unlike the three-stage training strategy that uses uncompressed video data as label, the proposed method uses higher-quality (lower QP setting) compressed data as label for HOP training. If the input is a compressed video frame with a large QP, the feature information contained in the compressed video frame is limited, which has too much gap from its ground truth (uncompressed video frames) to reconstruct. Moreover, uncompressed video frames do not contain compression artifacts so that the three-stage training strategy

is not effective in capturing the relationship between the compressed input and its uncompressed label. Based on the QP distance, the proposed method effectively captures the relationship between the compressed input and its label while successfully treating the QP balance problem.

- Unlike the three-stage training strategy that uses the compressed data generated by the original VTM and the HOP-embedded VTM, the proposed method adopts progressive learning based on QP distance that utilizes lower QP compressed data as labels for training, thus strengthening the HOP learning ability of removing compression artifacts. The proposed method achieves average BD rate gains of -8.31% (Y), -16.28% (U), and -18.27% (V) over the VTM-11.0 anchor in the AI configuration.

2 PROPOSED METHOD

2.1 Three-Stage Training Strategy by JVET

In the three-stage training strategy [8,9], the training set for Stage I is compressed by the original VTM, while the training sets for Stage II and Stage III are compressed by the HOP-embedded VTM (i.e. the previous stage model) as follows:

Stage I: 1) Extract a dataset of intra coded frames using VTM. 2) Train HOP on the dataset, resulting in the first model HOP I.

Stage II: 1) Extract a dataset of frames using VTM and the HOP model from Stage I. 2) Train HOP on the dataset, resulting in the second model HOP II. Integerized the model.

Stage III: 1) Extract a dataset of frames using VTM and the HOP model from Stage II. 2) Train HOP on the dataset, resulting in the final model HOP III.

2.2 QP Distance-Based Progressive Learning

HOP in-loop filter in JVET aims to achieve better compression artifact removal than the in-loop filters in the original VTM. In the three-stage training strategy recommended by JVET [8,9], the training sets in Stage II and Stage III are compressed more by the HOP-embedded VTM, i.e. 'VTM+CNN' in Fig. 2. Thus, the three-stage training strategy inevitably causes huge amount of training time. The CNN filters aim to remove compression artifacts from the predicted frames. In general, the CNN filters use uncompressed video frames, typically derived from the original video sequence, as label for training. As the original video sequence undergoes several processes such as intra prediction, inter prediction, transform and quantization, entropy coding, it loses substantial information in the original video frames. However, the uncompressed video frames do not contain compression artifacts and thus direct use of the uncompressed video frames as label for training is not effective in capturing the relationship between the compressed input and its corresponding uncompressed label. Moreover, if the input is a compressed frame at large QP, the feature information contained in the frame

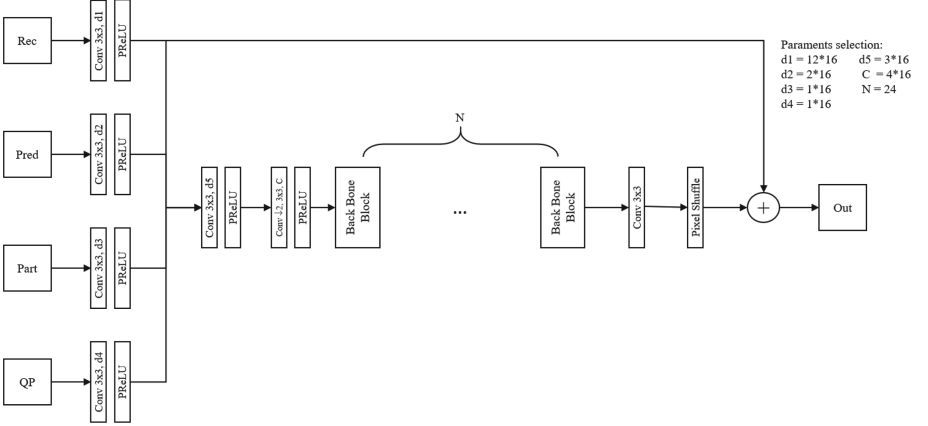


Fig. 4. Network architecture of HOP in-loop filter for luma component. Rec: Reconstructed frame. Pred: Predicted frame. Part: Partition map. QP: QP map. Out: Output frame. The luma channel has rich textures and uses more residual block extraction structures.

is very limited due to the severe feature loss. It causes much gap from its label, i.e. uncompressed video frames, and thus there exists QP balance problem in training. In this work, we propose to use higher-quality (lower QP setting) compressed frames as label for training rather than using the uncompressed frames as label. We introduce the QP distance into network training to set the label for the compressed input and adopt progressive learning based on QP distance to strengthen the HOP learning ability. The proposed training strategy is illustrated in Fig. 3. The training sets for all steps are derived from the original VTM compression (with QP settings of 7, 12, 17, 22, 27, 32, 37, 42). When setting the QP distance to 5, the input QP values for the training set are designated as 22, 27, 32, 37, 42, and the corresponding label QP values are set as 17, 22, 27, 32, 37. Subsequently, in the next training step, the model from the previous step is loaded, and the QP distance for the training set is increased. We conduct a total of four training steps, with QP distances of 5, 10, and 15 for the first three steps. The label QP used in the final step is set to 7. It is unnecessary to compress the training set with VTM at each step.

3 EXPERIMENTAL RESULTS

For experiments, we separately train the HOP model for the Y (Luma) and UV (Chroma) channels. The specific structures and parameter selection are illustrated in Figs. 4 and 5. According to the Common Test Conditions (CTC), the HOP model trained by the proposed training strategy is evaluated and compared with VTM-11.0 [11].

Table 1. BD-rate of Model I over VTM-11.0 in AI configuration. Model I is embedded into VTM-11.0_NNVC-2.0 [11] for evaluation. The QP distance is set to 5, the input QP values for the training set are designated as 22, 27, 32, 37, 42, while the corresponding label QP values are set as 17, 22, 27, 32, 37.

	Y-PSNR	U-PSNR	V-PSNR
Class A1	-5.99%	-15.29%	-17.12%
Class A2	-6.23%	-15.20%	-14.06%
Class B	-6.18%	-15.50%	-16.51%
Class C	-6.52%	-14.70%	-15.97%
Class E	-9.39%	-18.99%	-18.49%
Overall	-6.77%	-15.82%	-16.42%
Class D	-6.20%	-13.89%	-15.82%

Table 2. BD-rate of Model II over VTM-11.0 in AI configuration. Model II is embedded into VTM-11.0_NNVC-2.0 [11] for evaluation. The QP distance is set to 10, the input QP values for the training set are designated as 22, 27, 32, 37, 42, while the corresponding label QP values are set as 12, 17, 22, 27, 32.

	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.05%	-15.56%	-17.12%
Class A2	-7.22%	-17.18%	-17.11%
Class B	-7.19%	-15.00%	-16.62%
Class C	-7.69%	-15.92%	-17.91%
Class E	-10.67%	-20.92%	-21.53%
Overall	-7.86%	-16.65%	-17.89%
Class D	-7.45%	-15.41%	-17.73%

Table 3. BD-rate of Model III over VTM-11.0 in AI configuration. Model III is embedded into VTM-11.0_NNVC-2.0 [11] for evaluation. The QP distance is set to 15, the input QP values for the training set are designated as 22, 27, 32, 37, 42, while the corresponding label QP values are set as 7, 12, 17, 22, 27.

	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.38%	-15.57%	-17.07%
Class A2	-7.52%	-17.67%	-17.21%
Class B	-7.52%	-14.29%	-16.74%
Class C	-8.07%	-15.56%	-18.23%
Class E	-11.02%	-21.43%	-21.76%
Overall	-8.20%	-16.54%	-18.04%
Class D	-7.87%	-15.49%	-18.12%

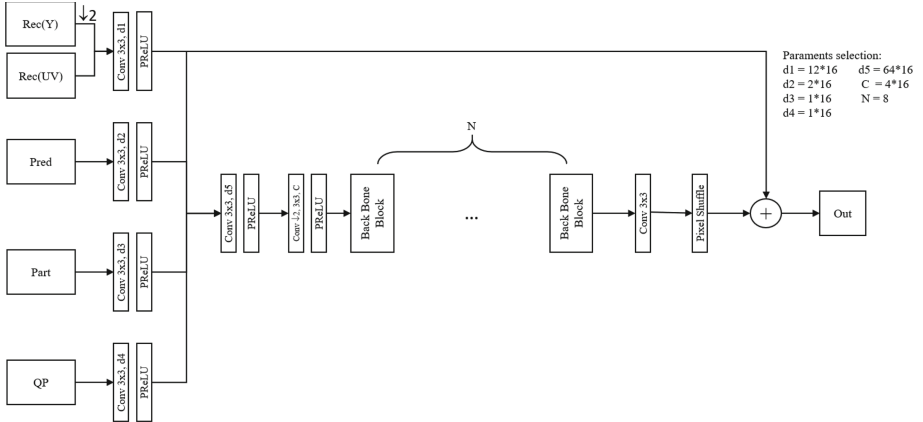


Fig. 5. Network architecture of HOP in-loop filter for chroma components. Rec: Reconstructed frame. Pred: Predicted frame. Part: Partition map. QP: QP map. Out: Output frame. The chroma channels have relatively less textures and is combined with the output frame of the luma channel Rec(Y) that is downsampled twice in the network input.

3.1 Experimental Setting

The experiments are conducted on PyTorch framework. and each step has 120 epochs. The loss function for the first 90 epochs is L1 loss, and the loss function for the last 30 epochs is L2 loss. The four steps total 480 epochs. The batch size is 32 and the learning rate is $1e-4$, decaying by half every 30 epochs. The DIV2K [1] and BVI-DVC [16] datasets are used to train HOP in-loop filter. All images are compressed using VTM-11.0 [11]. We randomly crop the compressed image into 144×144 patches and use random horizontal and vertical flipping for data augmentation. Then, we embed the trained CNN model into VTM-11.0_NNVC-2.0 [11] for evaluation. We use LibTorch to embed the HOP model in VTM, replacing DBF and SAO. In the test phase, we use the test sequences (A1, A2, B, C, D and E classes) in the Common Test Conditions (CTC) as the test set [3], and Bjøntegaard-Delta Bit-Rate (BD-BR) [5] as the evaluation metric to evaluate the performance of the proposed training strategy under AI configuration when training the HOP model.

3.2 Visual Comparison

We provide visual comparison of the proposed method with the three-stage training strategy [8,9] in Fig. 6. We obtain the decoded frames using two HOP models generated by the three-stage training strategy and the proposed method, when QP is 42. As highlighted in the zoomed areas, the proposed method generates better textures in images than the three-stage training strategy. This is mainly due to the progressive learning based on QP distance that uses lower QP compressed data as label for training, thus strengthening the HOP learning ability of removing compression artifacts. Moreover, the three-stage training

strategy utilizes uncompressed video frames as labels and is not effective in capturing the relationship between the compressed input and its uncompressed label. However, the proposed method effectively captures the relationship between the compressed input and its label based on the QP distance.

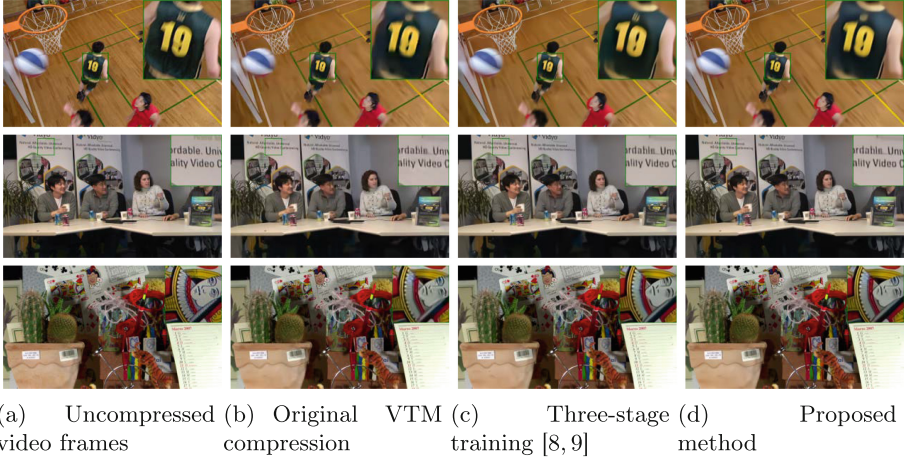


Fig. 6. Visual comparison between the three-stage training strategy [8,9] and the proposed method at QP=42.

Table 4. BD-rate of Model IV over VTM-11.0 in AI configuration. Model IV is embedded into VTM-11.0_NNVC-2.0 [11] for evaluation. The input QP values of the training set are specified as 22, 27, 32, 37, 42, while the corresponding label QP values are all set to 7.

	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.39%	-15.40%	-17.67%
Class A2	-7.63%	-18.03%	-17.31%
Class B	-7.62%	-13.98%	-16.98%
Class C	-8.25%	-15.46%	-18.50%
Class E	-11.14%	-20.37%	-21.69%
Overall	-8.31%	-16.28%	-18.27%
Class D	-8.05%	-15.35%	-17.96%

3.3 Performance Comparison and Training Time

Following CTC [3], we conducted the tests on the HOP model trained by the proposed method. In the AI configuration, the HOP model at the final stage

Table 5. BD-rate of the three-stage training strategy [8,9] over VTM-11.0 in AI configuration. The HOP model generated by the three-stage training strategy is embedded into VTM-11.0_NNVC-5.0 for evaluation.

	Y-PSNR	U-PSNR	V-PSNR
A1	-7.15%	-17.88%	-21.43%
A2	-7.09%	-19.27%	-17.10%
B	-7.09%	-18.05%	-19.85%
C	-8.11%	-18.42%	-21.04%
E	-10.56%	-20.36%	-21.71%
Overall	-7.91%	-18.69%	-20.23%
D	-7.95%	-18.14%	-21.60%

Table 6. Complexity comparison between the three-stage training strategy [8,9] and the proposed method in terms of the Multiply Accumulate (MAC)/pixel, the number of parameters and training time. The three-stage training strategy [8,9] is tested on Tesla V100 SXM2 32GB/Tesla A100 40GB GPU, while the proposed method is tested on a RTX 4090 32GB GPU.

	MAC/pixel	Number of parameters	Training time
Three-stage	477K	1.45M	45days
Proposed	371K (luma), 259K (chroma)	1.44M (luma), 0.76M (chroma)	10days

by the proposed method, i.e. Model IV, achieves average BD-rate reductions of -8.31%, -16.28%, and -18.27% over the VTM-11.0 anchor in the Y, U, and V channels, respectively. Tables 1, 2, 3 and 4 show the performance improvement of the HOP models at various steps (from Model I to Model IV) by the proposed method. In the Y channel, the proposed method gradually decreases BD-rate of -6.77%, -7.86%, -8.20%, and -8.31% from Model I to Model IV, respectively. This is because the progressive learning based on QP distance can strengthen the HOP learning ability of removing compression artifacts by using lower QP compressed data as labels. For reference, Table 5 provides BD-rate of the three-stage training strategy [8,9] over VTM-11.0 in AI configuration. The proposed method achieves comparable performance to the three-stage training strategy [8,9] and BD-rate reduction of 0.4% over it in the Y channel even with less training time. Table 6 provides complexity comparison between the three-stage training strategy and the proposed method in terms of the Multiply Accumulate (MAC)/pixel, the number of parameters and training time. The three-stage training strategy [8,9] is tested on Tesla V100 SXM2 32GB/Tesla A100 40GB GPU, while the proposed method is tested on a RTX 4090 32GB GPU. We each generate the HOP model for the Y (Luma) and UV (Chroma) channels as shown in Figs. 4 and 5, and provide them separately. Compared with the three-stage training strategy, the proposed method significantly reduces the training time from 45days to 10days.

4 CONCLUSION

In this paper, we have proposed QP distance-leveraged acceleration of HOP training to remarkably reduce its training time. The three-stage training strategy recommended by JVET used the HOP-embedded VTM to compress the training set twice more, thus causing huge amount of training time. To deal with this problem, we have presented progressive learning based on QP distance that does not use the training set compressed by the HOP-embedded VTM during the training process. Moreover, based on QP distance, we have used lower QP compressed data as label for training rather than using the uncompressed data as label. Thus, the proposed method effectively captures the relationship between the compressed input and its label, and strengthens the HOP learning ability of removing compression artifacts. Experimental results demonstrate that the proposed method achieves average BD-rate reductions of -8.31%, -16.28%, and -18.27% over the VTM-11.0 anchor in the Y, U, and V channels, respectively, and the proposed progressive learning strengthens the HOP learning ability of removing compression artifacts based on QP distance. Moreover, the proposed method significantly reduces the HOP training time.

Our future work includes extending the progressive learning strategy to various compression models for images and videos.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)
2. Alshina, E., Galpin, F.: BoG report on nn-filter design unification. Tech. Rep. JVET-AD0380, BoG (Apr 2023)
3. Alshina, E., Liao, R.L., Liu, S., Segall, A.: JVET common test conditions and evaluation procedures for neural network-based video coding technology. Tech. Rep. JVET-AE2016, JVET (Jul 2023)
4. Amruthavalli, P.L., Nalluri, P.: A review on in-loop filters for hevc and vvc video coding standards. In: Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS). vol. 1, pp. 997–1001 (2022). <https://doi.org/10.1109/ICACCS54159.2022.9784992>
5. Bjontegaard, G.: Calculation of average psnr differences between rd-curves. ITU-T VCEG-M33 (2001)
6. Chang, R., Wang, L., Xu, X., Liu, S.: Ee1-1.7: Optimization of training and network for nnvc filter set 0. Tech. Rep. JVET-AD0166, Tencent (Apr 2023)
7. Eadie, S., Li, Y., Rusanovskyy, D., Karczewicz, M.: Ee1-1.3: Reduced complexity cnn-based in-loop filtering. Tech. Rep. JVET-AD0205, Qualcomm (Apr 2023)
8. Galpin, F., Eadie, S., Li, Y., Wang, L., Xie, Z., Rusanovskyy, D., Li, Y., Chang, R., Li, J., Alshina, E.: Ahg11: Ee1-0 high operation point model. Tech. Rep. JVET-AE0191, InterDigital, Qualcomm, ByteDance, Tencent, Oppo, Huawei (Jul 2023)
9. Galpin, F., Li, Y., Wang, L., Rusanovskyy, D., Ström, J., Wang, L.: Description of algorithms and software in neural network-based video coding (nnvc) version 5. Tech. Rep. JVET-AF2019, InterDigital, Bytedance, Qualcomm, Ericsson, Tencent (Oct 2023)

10. Jia, W., Li, L., Li, Z., Zhang, X., Liu, S.: Residual-guided in-loop filter using convolution neural network. *ACM Transactions on Multimedia Computing, Communications, and Applications* **17**(4), 139.1–139.19 (2021)
11. Joint Video Experts Team (JVET): VTM-11.0-NNVC. https://vcgit.hhi.fraunhofer.de/jvet-ahg-nnvc/VVCSsoftware_VTM/-/tree/VTM-11.0_nnvc
12. Karczewicz, M., Hu, N., Taquet, J., Chen, C.Y., Misra, K., Andersson, K., Yin, P., Lu, T., François, E., Chen, J.: Vvc in-loop filters. *IEEE Trans. Circuits Syst. Video Technol.* **31**(10), 3907–3925 (2021)
13. Li, Y., Eadie, S., Rusanovskyy, D., Karczewicz, M.: Eel-related: Combination test of eel-1.3.5 and multi-scale component of eel-1.6. Tech. Rep. JVET-AD0211, Qualcomm (Apr 2023)
14. Li, Y., Zhang, K., Zhang, L., Eadie, S., Rusanovskyy, D., Karczewicz, M.: Eel-1.6: In-loop filter with wide activation and large receptive field. Tech. Rep. JVET-AD0106, Bytedance, Qualcomm (Apr 2023)
15. Li, Y., Zhang, L., Zhang, K.: Convolutional neural network based in-loop filter for vvc intra coding. In: *Proceedings of the IEEE Conference on Image Processing*. pp. 2104–2108 (2021)
16. Ma, D., Zhang, F., Bull, D.R.: BVI-DVC: A training database for deep video compression. *IEEE Trans. Multimedia* **24**, 3847–3858 (2021)
17. Shingala, J.N., Shyam, A., Suneja, A., Badya, S.P., Shao, T., Arora, A., Yin, P., Pu, F., Lu, T., McCarthy, S.: Eel-1.10: Complexity reduction on neural-network loop filter. Tech. Rep. JVET-AC0106, Ittiam Systems, Dolby Laboratories (Jan 2023)
18. Wang, H., Chen, J., Reuze, K., Kotra, A.M., Karczewicz, M.: Eel-related: Neural network-based in-loop filter with constrained computational complexity. Tech. Rep. JVET-W0131, Qualcomm Inc. (Jul 2021)
19. Wang, L., Jiang, W., Xu, X., Liu, S.: Eel-1.3: neural network based in-loop filter. Tech. Rep. JVET-X0052, Tencent (Oct 2021)
20. Wang, L., Xu, X., Liu, S., Galpin, F.: Eel-1.2: Neural network based in-loop filter with a single model. Tech. Rep. JVET-Z0091, Tencent, InterDigital (Apr 2022)
21. Wennersten, P., Ström, J., Liu, D.: Eel-1.4: Channel redistribution for luma and chroma. Tech. Rep. JVET-AD0168, Ericsson (Apr 2023)
22. Zhang, H., Jung, C., Liu, Y., Li, M.: Lightweight cnn-based in-loop filter for vvc intra coding. In: *Proceedings of the IEEE Conference on Image Processing*. pp. 1635–1639 (2023)
23. Zhang, Y., Wang, X., Dai, Q., Yan, C., Dai, F., Li, L.: Parallel deblocking filter for hevc on many-core processor. *Electron. Lett.* **50**(5), 367–368 (2014)



Asymmetric Learned Image Compression Using Fast Residual Channel Attention

Yusong Hu¹, Cheolkon Jung^{1(✉)}, Yang Liu², and Ming Li²

¹ School of Electronic Engineering, Xidian University, Xian 710071, China
zhengzk@xidian.edu.cn

² Guangdong OPPO Mobile Telecommunications Corporation, Dongguan, China

Abstract. In recent years, substantial advances have been made in deep learning-based image compression. Most studies have focused on designing accurate and flexible entropy models to predict the distribution of latent features in images. However, the allocation of computing resources and the restoration of decoded images by post-processing are equally important. In this paper, we propose asymmetric learned image compression based on fast residual channel attention. We design an asymmetric image compression network to effectively allocate computational resources into the post-processing of the decoder. Inspired by image super-resolution, we provide a fast residual channel attention module in the post-processing based on depthwise separable convolution. This module can quickly restore the features lost by compression, resulting in image quality enhancement. Experimental results demonstrate that the proposed method outperforms state-of-the-art methods for learned image compression in terms of PSNR, MS-SSIM and runtime.

Keywords: Image compression · convolutional neural network · depthwise separable convolution · fast residual channel attention · variational auto-encoder

1 Introduction

Image compression refers to the process of reducing the size of digital images in limited bandwidth while preserving their visual quality. It has been widely used for transmission, storage, and processing of digital images. By compressing images, people can save storage space, reduce transmission bandwidth, and speed up image processing. In recent decades, a plenty of lossy image compression methods have been developed, which are broadly categorized into two types: traditional and deep learning-based. Traditional methods, including JPEG [32], JPEG2000 [9], BPG [6], and VVC [7], achieve a high compression rate but reach the limit of improvement in compression efficiency. There still exist compression-related issues, such as block distortion and mosaic artifacts. In recent years,

This work was supported by the National Natural Science Foundation of China (No. 62111540272).

researchers have turned to investigate introducing deep learning into image compression, called deep learning-based image compression or learned image compression.

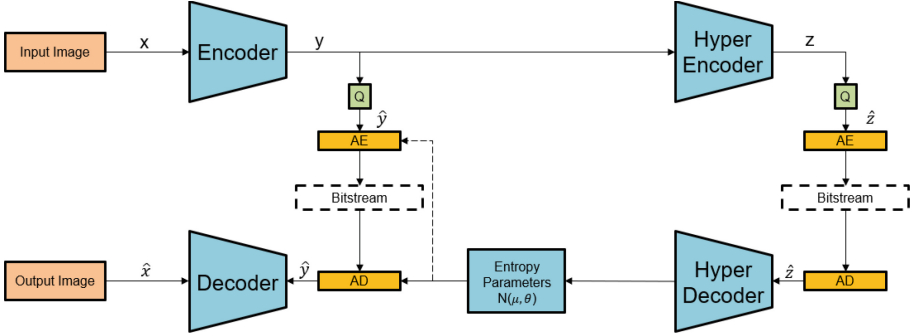


Fig. 1. Overall process of the VAE-based image compression. Q: Quantization. AE: Arithmetic encoder. AD: Arithmetic decoder.

Deep learning-based image compression has gained significant attention by researchers. It mainly utilizes convolutional neural networks (CNNs) for end-to-end learning while effectively reducing distortion during compression while maintaining a high compression rate. The variational autoencoder (VAE) is the most notable method for end-to-end learned image compression. Fig. 1 illustrates the basic flow of VAE for end-to-end learned image compression. For encoding, VAE-based image compression adopts a combination of linear and nonlinear parametric transforms to map an image to a latent space. After quantization, entropy estimation modules predict the latent distribution, then are compressed into a bitstream using the lossless context-based adaptive binary arithmetic coding (CABAC) or range coder (RC). Additionally, hyper-prior, auto-regressive priors, and Gaussian Mixture Model (GMM) enable the entropy estimation modules to more accurately predict the distributions of the latents, leading to better Rate-Distortion (RD) performance. For decoding, the lossless CABAC or RC decompresses the bitstream, and the decompressed latents are mapped to reconstructed images using a linear and nonlinear parametric synthesis transform.

In this paper, we propose an asymmetric learned image compression network using fast residual channel attention. The proposed method adopts an asymmetric structure between encoder and decoder to generate a concise bitstream using simple encoding and reconstruct the output image through complex decoding. This approach enables better image recovery while maintaining a high compression rate. The encoder minimizes bits for transmission, while the decoder performs post-processing for reconstruction based on depth separable convolution to be faster with fewer parameters. The encoder employs a straightforward

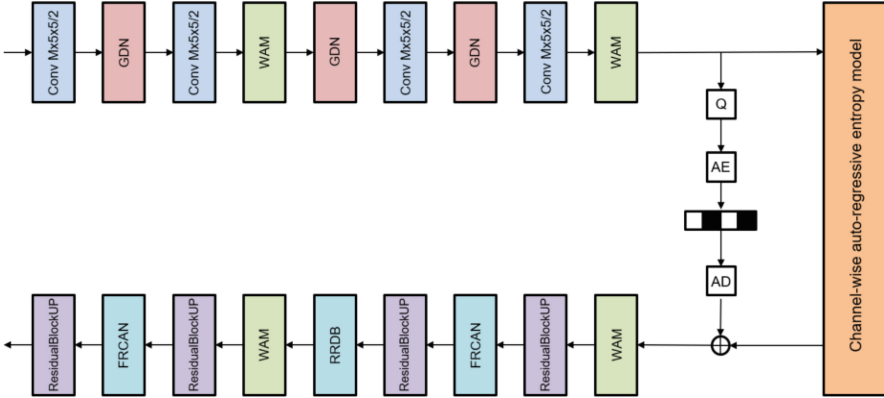


Fig. 2. Entire framework of the proposed learned image compression. GDN: generalized divisive normalization. WAM: Window attention module. FRCAN: Fast residual channel attention network. RRDB: Residual in residual dense block. Q: Quantization. AE: Arithmetic encoder. AD: Arithmetic decoder. In the encoder, we use 5×5 down sampling convolution, GDN [2] and WAM [39], while in the decoder we apply a large number of residual structure networks and attention modules to recover the lost features. For the entropy coding, we utilize the Minnen and Singh’s method [26].

design to capture visual characteristics, resulting in data size reduction and compression rate improvement. We use 5×5 down-sampling convolution, generalized divisive normalization (GDN) [2] and window attention module (WAM) [39] in the encoder. The decoder incorporates residual networks with attention mechanisms to enhance image clarity and restore structural details. The combination of residual learning and channel attention in the decoder restores image features, resulting in effective image recovery. For the entropy coding, we utilize the Minnen and Singh’s method [26] to achieve good rate-distortion performance. Experimental results demonstrate that the proposed method outperforms existing state-of-the-art image compression methods, exhibiting superior performance in terms of both PSNR and MS-SSIM metrics, especially at medium to high bit-rates. Moreover, the proposed network maintains outstanding efficiency in the encoding and decoding speed, thereby highlighting its practical significance. Fig. 2 illustrates the entire framework of the proposed learned image compression.

Compared with existing methods, main contributions of this paper are summarized as follows:

- We propose an asymmetric image compression network based on VAE that achieves fast decoding speed with a high compression rate. The encoder adopts a simple structure to extract image features, thus resulting in a reduced data stream. Meanwhile, the decoder utilizes residual networks and attention mechanisms to recover details and structures in an image. Although the

decoder contains several residual networks, it maintains fast decoding speed by depthwise separable convolution.

- Inspired by image super-resolution, we present an image restoration module that combines dense residual networks and channel attention mechanisms. Dense residual networks improve the quality of image restoration, while channel attention mechanisms select features selectively, thus leading to further performance improvement.
- We use depthwise separable convolution to replace the convolution layers in the decoder, thus accelerating decoding speed. To our knowledge, this is the first application of depthwise separable convolution in deep learning-based image compression task.

2 Related Work

2.1 Learned Image Compression

In 2016, Ballé et al. [3] presented a novel deep learning-based image compression method, which first incorporates end-to-end learning with image compression. This method outperforms traditional compression methods in terms of both compression rate and image quality. Moreover, it demonstrates superior efficiency and capability to handle a larger image size. Afterward, Ballé et al. [4] further proposed an image compression method based on their previous work and introduced scale hyperprior to enhance the compression performance. They achieve more efficient encoding and decoding while maintaining compression rate and image quality. Several methods [11, 28] utilized generative models and trained adversarially to learn the image distribution for subjective quality at a low bitrate. Jiang et al. [12] introduced super-resolution into image compression to save bits. Li et al. [19, 20] proposed content weighted image compression for spatial transformation and quantization representation based on deep learning. In 2020, Cheng et al. [8] proposed learned image compression based on discretized Gaussian mixture likelihoods and attention. This is the first deep learning method to achieve comparable performance to VVC intra coding [7]. Zou et al. [39] incorporated a window attention mechanism (WAM) to take correlations between adjacent elements in space, resulting in the performance improvement in image compression.

2.2 Attention Mechanism

Attention mechanisms have become a fundamental concept in neural networks, which have numerous applications such as natural language processing, statistical learning, speech recognition, and computer vision. Wang et al. [33] proposed a novel neural network model, which introduces the concept of non-local blocks to capture long-range dependencies between pixels through computing and weighted averaging of global features. Woo et al. [36] proposed a new modular attention mechanism, called Convolutional Block Attention Module (CBAM),

which can be embedded into CNN models to further enhance their performance. Liu et al. [23] proposed an image compression method based on attention mechanism and non-local operation, aiming to improve the visual quality of compressed images. They used non-local operations to generate attention maps, allocating more bits to important regions for adaptive processing of latent features, thereby achieving better image compression performance. Recently, transformer has been introduced into the learned image compression. Liu et al. [24] proposed a parallel Transformer-CNN Mixture (TCM) block with a controllable complexity to combine the local learning ability of CNN and the non-local learning ability of transformers.

2.3 Image Super-Resolution

Image Super-resolution (SR) aims to reconstruct high-quality images using only low-quality input information. Since the pioneering work of Super-Resolution Convolutional Neural Network (SR-CNN) [10], deep learning-based methods have dominated the field of image super-resolution. Zhang et al. [38] proposed a method for image super-resolution using very deep residual channel attention networks. This method combines residual connections and channel attention mechanisms, while utilizing a very deep neural network architecture. SwinIR [22] is an image restoration method based on swin transformer proposed by Li et al. in 2021. It uses swin transformer to perform downsampling and upsampling on images, while preserving more information with residual connections. It has been reported that SwinIR achieves excellent performance on multiple image restoration tasks, while maintaining high computational efficiency SwinIR [22].

3 Proposed Method

3.1 Network Architecture

Image compression aims to achieve a high compression rate while maintaining the quality of the reconstructed images, i.e. use less bitstream to recover a better decompressed image than the anchor. A possible approach to the end-to-end image compression is to use simple encoding ends for feature extraction and complex decoding ends for image reconstruction, thus resulting in small bitstream. In this work, we attempt to enlarge the complexity difference between the encoding and decoding ends and design an asymmetric network architecture to obtain a high compression rate as shown in Fig. 2. In the encoding end, we use simple convolutional layers and windowed attention mechanisms to downsample the input image and perform feature selection. In contrast, we employ a large number of residual structures in the decoding end to recover the high-frequency information lost during the downsampling and entropy coding processes. We also use attention modules to select appropriate image features to maintain the quality of the reconstructed images. In addition, encoding and decoding time is a critical factor for evaluating the performance of an image compression algorithm.

Therefore, we utilize the minimum-serial processing entropy coding structure proposed by Minnen and Singh [26] for fast entropy coding. Furthermore, the complex decoding structure of the proposed network inevitably leads to longer decoding time. To address this issue, we optimize the decoding process based on depthwise separable convolution.

3.2 Depthwise Separable Convolution

To decrease the computational and storage complexity of the dense residual network, depthwise separable convolution (DSC) is utilized instead of regular convolution. The principle of DSC is to decompose the traditional convolution kernel into two independent convolution kernels, thereby reducing the number of parameters required in convolution. Specifically, in the regular convolution operation, the convolution kernel contains parameters in the height and width directions, while performing convolution operations on all input channels. In DSC, convolution is decomposed into two steps: depthwise convolution and pointwise convolution. The input feature map is initially divided into single-channel feature maps along the depth dimension, followed by the application of a depthwise convolution kernel with a size of either 3×3 or 5×5 to extract channel-specific features from each single-channel feature map. Subsequently, a pointwise convolution using a 1×1 kernel is used to combine the features across different channels. It allows the depthwise convolution kernel to only include parameters in the height and width directions, while the pointwise convolution kernel only includes parameters in the channel direction. Since DSC separates the convolution operation into two independent operations, it can significantly reduce the number of parameters and calculations required while maintaining the same performance. This, in turn, reduces both training and inference time. Therefore, we replace the ordinary convolution layer in the residual in residual dense block (RRDB) [34] and the proposed fast residual channel attention network (FRCAN) with DSC to minimize computational complexity and improve the processing speed. Since DSC remarkably reduces the number of parameters, it can reduce the complexity of the decoder.

3.3 Fast Residual Channel Attention Network

In image super-resolution, it has been proven that the Residual Channel Attention Network (RCAN) proposed by Zhang et al. [38] can increase the depth and receptive field of the network and improve its performance by adding a large number of residual blocks and attention modules between the input and output layers. Since the decoding flow in image compression is similar to image super-resolution, inspired by Zhang et al.'s work, we design a fast residual channel attention network (FRCAN) and incorporate it into the decoding end. As shown in Fig. 3, we present a novel module, called residual channel attention block (RCAB), which incorporates a simplified version of the dense residual network prior to the channel attention mechanism to enhance the recovery of image

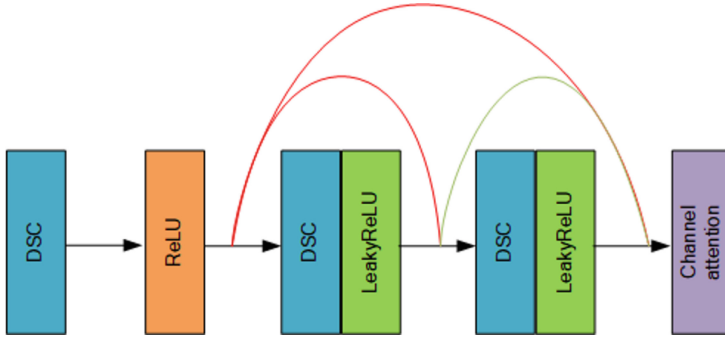


Fig. 3. Network structure of the proposed residual channel attention block (RCAB). DSC: Depthwise separable convolution.

features. The channel attention mechanism then filters and selects the most relevant features from the retrieved channel features. We combine four RCAB modules to form FRCAN, which is used to generate and select appropriate channel features.

3.4 Residual in Residual Dense Block

As the residual structure in FRCAN is relatively simple, it cannot generate more features that are lost during image compression progress. Therefore, we introduce an improved RRDB module [34] at the decoding end to provide more features. As shown in Fig. 4, we also replace the convolution layers in the RRDB module with DSC to reduce computational complexity.

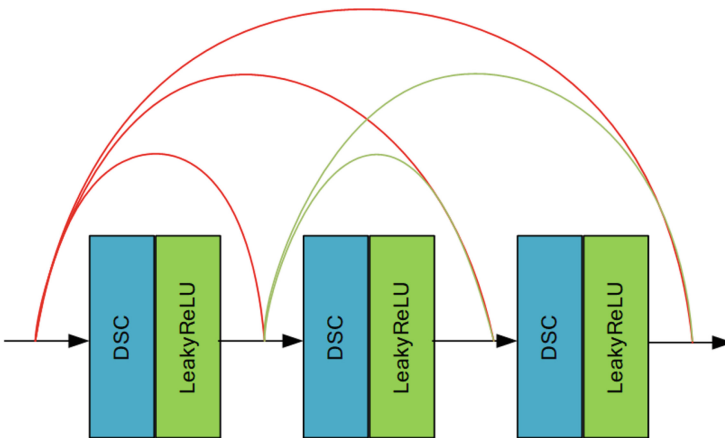


Fig. 4. Network architecture of the proposed residual in residual dense block (RRDB). DSC: Depthwise separable convolution.

3.5 Loss function

In image compression, we employ an encoder E to transform the original image x into a latent representation y . The quantization operation Q discretizes y into \hat{y} , which is subsequently used by a decoder D to reconstruct the image as \hat{x} as follows:

$$\begin{aligned} y &= E(x; \phi) \\ \hat{y} &= Q(y) \\ \hat{x} &= D(\hat{y}; \theta) \end{aligned} \quad (1)$$

where ϕ and θ are the trainable parameters of the encoder E and decoder D .

Since quantization Q inevitably introduces truncation errors to the latent representation, which can cause distortion in reconstructed images. Therefore, in the training phase, we follow the Minnen and Singh’s method [26] by correcting the quantization error through rounding and adding the predicted quantization error. We model each element \hat{y}_i as a single Gaussian distribution with its standard deviation σ_i and mean μ_i , and introduce side information \hat{z}_i to generate the distribution $p_{\hat{y}_i|\hat{z}_i}$. This distribution is modeled using an SGM-based entropy model, which is formulated as follows:

$$p_{\hat{y}_i|\hat{z}_i}(\hat{y}_i|\hat{z}_i) = N(\mu_i, \sigma_i^2) \quad (2)$$

The loss function of the proposed network is controlled by the rate-distortion trade-off term R and D , which is expressed as follows:

$$\begin{aligned} L &= R + \lambda * D \\ &= E_{x \sim px} [-\log_2 p_{\hat{y}_i|\hat{z}_i}(\hat{y}_i|\hat{z}_i) - \log_2 p_{\hat{z}_i}(\hat{z}_i)] \\ &\quad + \lambda \cdot E_{x \sim px} [d(x, \hat{x})] \end{aligned} \quad (3)$$

where λ controls the trade-off between rate and distortion, R is the bit-rate of latents \hat{y} and \hat{z} , $d(x, \hat{x})$ is the distortion between the uncompressed image x and the reconstructed image \hat{x} .

4 Experiments

4.1 Experimental Setup

Training: We train the proposed image compression framework with different λ values ($\lambda = 0.0016, 0.0032, 0.0075, 0.015, 0.03, 0.045$) using the CompressAI platform [5]. For training, we randomly choose 300k images from the OpenImages dataset [17], and randomly crop them with the size of 256×256 . All models are trained for 1.45M steps using the Adam optimizer [15] with a batch size of 16. The initial learning rate is set to 1×10^{-4} for 900k iterations, which drops to 3×10^{-5} for another 250k iterations and 1×10^{-5} for the last 300k iterations.

Evaluation: We assess the effectiveness of the proposed method by testing it on Kodak24 image dataset [16] and JPEG AI testing dataset [14]. We follow the requirements of the JPEG AI Common Testing Conditions (CTC) [13] released by the JPEG AI Ad Hoc Group (AhG) in January 2022, and use VTM-11.1 as the anchor for VVC.



Fig. 5. Visual quality comparison on the JPEG AI testing dataset. We compare the results of the proposed method with Ballé et al. [4], Minnen et al. [25], Cheng et al. [8], VTM-11.1 [13] and Zou et al. [39].

Table 1. BD-rate gains of different methods over the anchor VVC intra coding (VTM-11.1) [7] on Kodak dataset. A positive number indicates that the performance is worse than the anchor, while a negative number indicates that the performance is better.

	VVC [7]	Zou [39]	Song [30]	Cheng [8]	Minnen [25]	Ballé [4]	JPEG2000 [9]	JPEG [32]	Proposed
PSNR	0	-7.71%	6.63%	1.23%	6.70%	28.39%	96.91%	229.70%	-8.97%
MS-SSIM	0	-10.86%	3.47%	-7.79%	-1.20%	15.69%	118.62%	180.62%	-11.08%

4.2 Performance Comparison

Visual Comparison: Fig. 5 provides visual quality comparison based on the JPEG AI testing dataset [14]. The proposed method outperforms Ballé et al.’s [4] method by utilizing fewer bits and yielding better PSNR and MS-SSIM. In comparison with Cheng et al.’s [8], Zou et al.’s [39], Minnen et al.’s [25] and VTM-11.1 [13], the proposed method employs a similar number of bits, yet produces higher PSNR and MS-SSIM values for the reconstructed image. Meanwhile, the proposed method outperforms other image compression methods in terms of clarity, naturalness, and achieving better detail features as demonstrated in the zoomed-in regions.

Quantitative Measurements: We have conducted a quantitative comparison of the proposed method with learned image compression techniques including Zou et al.’s [39], Song et al.’s [30], Cheng et al.’s [8], Minnen et al.’s [25], and Ballé et al.’s [4] as well as some traditional compression algorithms such as VVC (VTM-11.1) [7], JPEG [32], and JPEG2000 [9]. Figs. 6 and 7 show the rate-distortion performance comparison based on the Kodak24 dataset [16] using mean squared error (MSE) as the loss function for training. When using PSNR

Table 2. VVC reporting template. We use the proposed method as the anchor and compare it with various traditional image compression methods. To meet the requirements of the JPEG AI Ad Hoc Group (AhG) [13], we evaluate various indicators and record the results for presentation.

Method	AVG BD-rate	MS-SSIM	torch VIF [29]	FSIM [37]	NLPD [18]	IW-SSIM [35]	VMAF [21]	psnrHVS [27]
Proposed	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
JPEG2000 [9]	54.9%	66.0%	69.4%	49.8%	54.5%	62.7%	24.0%	58.2%
JPEGXL [1]	56.3%	46.9%	64.5%	26.3%	51.1%	60.7%	77.7%	67.2%
HEVC [31]	3.7%	2.3%	5.8%	15.7%	-0.8%	4.0%	0.8%	-2.2%
VTM-11.1 [13]	-7.2%	-6.2%	-6.4%	-4.0%	-9.6%	-5.8%	-10.2%	-8.3%

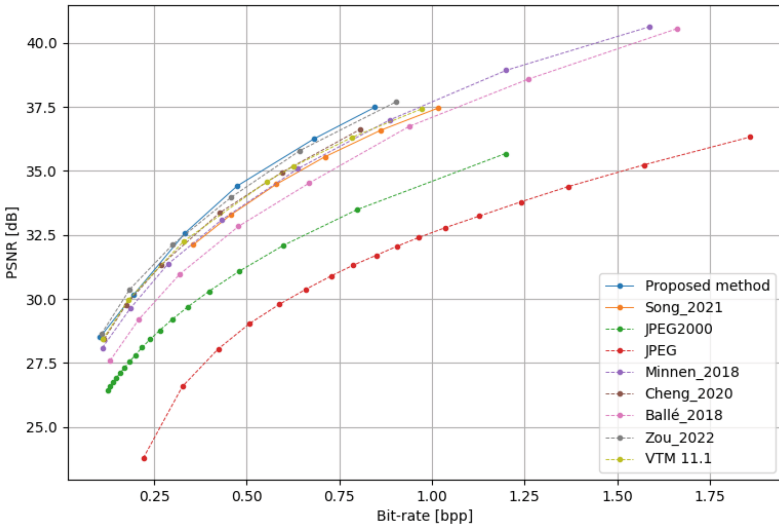


Fig. 6. RD curves among different methods on the Kodak24 dataset [16]: bitrate (bpp) versus PSNR (dB).

Table 3. Comparison of average encoding and decoding time among Ballé et al. [4], Minnen et al. [25], Cheng et al. [8] and Zou et al. [39] on Kodak dataset using one RTX 3090 GPU. Note that Cheng et al.’s results are based on a lightweight implementation (without Gaussian mixture likelihoods) in CompressAI framework [5].

Method	Enc(s)	Dec(s)	PSNR(dB)	MS-SSIM	bpp
Ballé [4]	0.0250	0.0189	34.53	0.9836	0.669
Minnen [25]	2.5202	5.3006	35.09	0.9837	0.639
Cheng [8]	3.0139	5.7965	34.95	0.9838	0.595
Zou [39]	0.0884	0.0916	35.80	0.9855	0.644
Proposed	0.0835	0.0979	36.24	0.9873	0.681

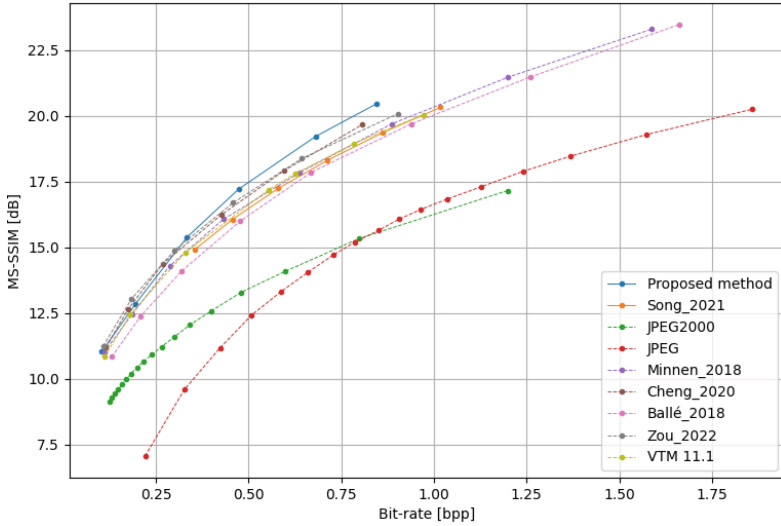


Fig. 7. RD curves among different methods on the Kodak24 dataset [16]: bitrate (bpp) versus MS-SSIM.



Fig. 8. Visual quality comparison on the Kodak24 dataset. We perform the ablation experiments on the FRCAN and RRDB modules in the proposed network. The FRCAN and RRDB modules effectively restores the image features lost by compression.

and MS-SSIM as the evaluation metric, the proposed method is comparable to the results of VTM-11.1 [13] and Cheng et al.’s at a low bitrate, and slightly inferior to Zou et al.’s. At medium to high bitrates, our results are superior to those of VTM-11.1, Cheng et al.’s, and Zou et al.’s. The proposed network performs slightly worse at a low bitrate. The reconstructed images at a low bitrate often contain a large amount of erroneous high-frequency information, which is further enhanced and selected by the FRCAN module at decoding end, thus leading to poor performance of the proposed network. Table 1 presents the BD rate performance while using VVC (VTM-11.1) as an anchor. The proposed method reaches approximately 8.97% and 11.08% rate gains in PSNR and MS-SSIM, respectively. Compared to other deep learning-based image compression methods, the proposed method obtains BD rate reduction of 1.26% to 37.36% in

PSNR evaluation and 0.22% to 26.77% in MS-SSIM evaluation. In addition, we compare the proposed method with traditional image compression methods such as JPEG and JPEG2000. The proposed method achieves rate improvements of 238.67% and 105.88% in PSNR evaluation, and 191.70% and 129.7% in MS-SSIM evaluation, respectively. According to the requirements of the JPEG AI Ad Hoc Group (AhG), we conduct tests on the JPEG AI testing dataset [14] using multiple objective evaluation metrics. The test results are shown in Table 2. Although the proposed method slightly outperforms HEVC [31], it has certain shortcomings when compared to VTM-11.1. In particular, we observe a discrepancy between the results obtained from the Kodak24 dataset and those from the JPEG AI testing dataset. This is attributed to the difference in resolution between the OpenImage training dataset, which is similar to the Kodak24 test dataset, however the high-resolution images (4K, 8K) in the JPEG AI testing dataset is unable to compress effectively by the proposed method.

Compression Efficiency: In Table 3, we compare the encoding and decoding speed of the proposed method with other deep learning-based methods on the Kodak24 testing dataset. The Kodak 24 test set contains 24 images, and each image has a different size. Thus, the average value in the table means the average time of encoding and decoding for each image. Due to a more complex decoding stage in the proposed network, the decoding speed is 0.0144 seconds slower than the encoding speed. Although a large number of residual structures are used in the decoding stage, DSC accelerates operation speed, enabling the proposed method to maintain the performance. This demonstrates outstanding contribution of DSC to the complexity of the proposed network. Compared to other methods, the proposed method performs better in both objective evaluation metrics and encoding/decoding time.

4.3 Ablation Study

To evaluate the contribution of the proposed FRCAN and RRDB modules to the image compression efficiency, we conduct an ablation experiment. Specifically, we remove FRCAN and RRDB modules from the decoder and train the model at four different λ values. The results are shown in Figs. 8 and 9. We conduct a visual comparison on the ablation experiment. As shown in Fig. 8, FRCAN and RRDB modules can recover better image details, thus further confirming the effectiveness of the proposed method for image compression. In addition, Fig. 9 indicates that integrating FRCAN and RRDB modules effectively restores the lost image features caused by compression, which leads to higher objective quality with a fewer bit-rate.

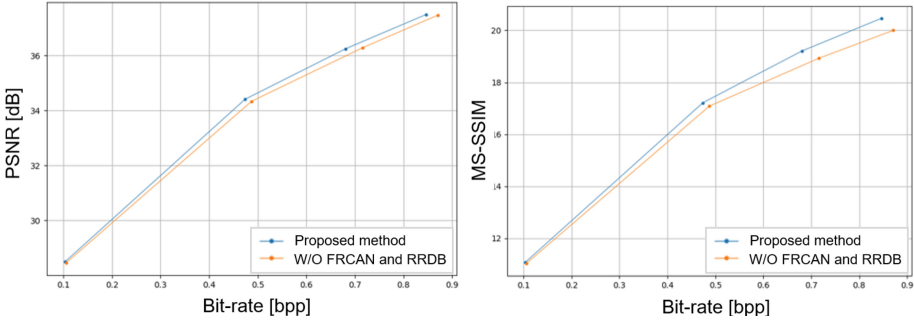


Fig. 9. Ablation experiments on the FRCAN and RRDB modules. Left: Bitrate (bpp) versus PSNR. Right: Bitrate (bpp) versus MS-SSIM.

5 Conclusion

In this paper, we have proposed a learned image compression network based on fast residual channel attention. We have presented a network that employs asymmetric structure between encoding and decoding to generate a concise bit-stream using a simple encoding end and recover the output image through a complex decoding end. This approach enables better image recovery while maintaining a high compression rate. Inspired by image super-resolution, we have introduced a fast residual channel attention network (FRCAN) based on DSC into the decoding end of the proposed network to quickly generate and select image features. Experimental results on Kodak24 dataset show that the proposed network achieves 8.97% and 11.08% gains over VVC in terms of PSNR and MS-SSIM, respectively. Furthermore, the proposed network provides a good balance between coding/decoding speed and visual quality.

Our future work involves joint learning of spatial and frequency features in the proposed network to further improve the compression efficiency.

References

1. Alakuijala, J., Asseldonk, R.V., Boukourt, S., Szabadka, Z., Wassenberg, J.: Jpeg xl next-generation image compression architecture and coding tools. In: Applications of Digital Image Processing XLII (2019)
2. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. arXiv preprint [arXiv:1511.06281](https://arxiv.org/abs/1511.06281) (2015)
3. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. arXiv preprint [arXiv:1611.01704](https://arxiv.org/abs/1611.01704) (2016)
4. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior. arXiv preprint [arXiv:1802.01436](https://arxiv.org/abs/1802.01436) (2018)
5. Bégaint, J., Racapé, F., Feltman, S., Pushparaja, A.: Compressai: a pytorch library and evaluation platform for end-to-end compression research. arXiv preprint [arXiv:2011.03029](https://arxiv.org/abs/2011.03029) (2020)

6. Bellard, F.: Bpg image format (2016), <https://bellard.org/bpg/>, Last accessed on 2018-4-21
7. Bross, B., Wang, Y.K., Ye, Y., Liu, S., Chen, J., Sullivan, G.J., Ohm, J.R.: Overview of the versatile video coding (vvc) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.* **31**(10), 3736–3764 (2021)
8. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: *Proc. IEEE CVPR*. pp. 7939–7948 (2020)
9. Christopoulos, C., Skodras, A., Ebrahimi, T.: The jpeg2000 still image coding system: an overview. *IEEE Trans. Consum. Electron.* **46**(4), 1103–1127 (2000)
10. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: *Proc. ECCV*. pp. 184–199 (2014)
11. Iwai, S., Miyazaki, T., Sugaya, Y., Omachi, S.: Fidelity-controllable extreme image compression with generative adversarial networks. In: *Proc. ICPR*. pp. 8235–8242 (2021)
12. Jiang, F., Tao, W., Liu, S., Ren, J., Guo, X., Zhao, D.: An end-to-end compression framework based on convolutional neural networks. *IEEE Trans. Circuits Syst. Video Technol.* **28**(10), 3007–3018 (2017)
13. JPEG: Iso/iec jtc 1/sc29/wg1 n100106, icq "jpeg ai common training and test conditions". URL <https://jpeg.org/jpegai/documentation.html> (2022)
14. JPEG: Jpeg ai common training and test conditions. URL <https://jpeg.org/jpegai/dataset.html> (2022)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
16. Kodak, E.: Kodak lossless true color image suite (photocd pcd0992). URL <http://r0k.us/graphics/kodak> (1993)
17. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from <https://github.com/openimages> **2**(3), 18 (2017)
18. Laparra, V., Ballé, J., Berardino, A., Simoncelli, E.P.: Perceptual image quality assessment using a normalized laplacian pyramid. In: *Proceedings of the Human Vision and Electronic Imaging*. pp. 43–48 (2016)
19. Li, M., Zuo, W., Gu, S., You, J., Zhang, D.: Learning content-weighted deep image compression. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(10), 3446–3461 (2020)
20. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. In: *Proc. IEEE CVPR*. pp. 3214–3223 (2018)
21. Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., Manohara, M.: Toward a practical perceptual video quality metric. *The Netflix Tech Blog* **6**(2), 2 (2016)
22. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: *Proc. IEEE ICCV*. pp. 1833–1844 (2021)
23. Liu, H., Chen, T., Guo, P., Shen, Q., Cao, X., Wang, Y., Ma, Z.: Non-local attention optimized deep image compression. arXiv preprint [arXiv:1904.09757](https://arxiv.org/abs/1904.09757) (2019)
24. Liu, J., Sun, H., Kattok, J.: Learned image compression with mixed transformer-cnn architectures. In: *Proc. IEEE CVPR*. pp. 14388–14397 (2023)
25. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. *Advances in Neural Information Processing Systems* **31** (2018)
26. Minnen, D., Singh, S.: Channel-wise autoregressive entropy models for learned image compression. In: *Proc. IEEE ICIP*. pp. 3339–3343 (2020)

27. Ponomarenko, N., Silvestri, F., Egiazarian, K., Carli, M., Astola, J., Lukin, V.: On between-coefficient contrast masking of dct basis functions. In: Proceedings of the International Workshop on Video Processing and Quality Metrics. vol. 4 (2007)
28. Santurkar, S., Budden, D., Shavit, N.: Generative compression. In: Proc. PCS. pp. 258–262 (2018)
29. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. *IEEE Trans. Image Process.* **15**(2), 430–444 (2006)
30. Song, M., Choi, J., Han, B.: Variable-rate deep image compression through spatially-adaptive feature transform. In: Proc. IEEE ICCV. pp. 2380–2389 (2021)
31. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
32. Wallace, G.K.: The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics* **38**(1), xviii–xxxiv (1992)
33. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proc. IEEE CVPR. pp. 7794–7803 (2018)
34. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proc. ECCV Workshops. pp. 0–0 (2018)
35. Wang, Z., Li, Q.: Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Process.* **20**(5), 1185–1198 (2010)
36. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proc. ECCV. pp. 3–19 (2018)
37. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**(8), 2378–2386 (2011)
38. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proc. ECCV. pp. 286–301 (2018)
39. Zou, R., Song, C., Zhang, Z.: The devil is in the details: Window-based attention for image compression. In: Proc. IEEE CVPR. pp. 17492–17501 (2022)



LDINet: Long Distance Imaging Through RGB and NIR Image Fusion

Lin Mei, Hao Zhang, and Cheolkon Jung^(✉)

School of Electronic Engineering, Xidian University, Xian 710071, China
zhengzk@xidian.edu.cn

Abstract. The fusion of visible color (RGB) and near infrared (NIR) images takes multispectral advantage of colors from RGB image and details from NIR image. Unlike RGB images, NIR images are robust to atmospheric environments such as Rayleigh scattering and Mie scattering. In this paper, we propose long distance imaging through RGB and NIR image fusion, named LDINet. We achieve hidden texture recovery for long distance imaging based on the fusion of RGB and NIR images. We adopt pyramid feature selection to capture multiscale information in the fusion network. Since overexposure and underexposure cause a dynamic range allocation problem in RGB image, we use the attention map of RGB image to adjust contrast enhancement. We synthesize the input smoothed RGB images for training by smoothing their original RGB images, i.e. ground truth. During training, we feed the smoothed RGB images and the details of NIR images into the fusion network as input, while feeding the ground truth as output. Experimental results show that LDINet successfully recovers hidden textures lost in RGB images while keeping colors and outperforms state-of-the-art fusion methods in terms of visual quality and quantitative measurements.

Keywords: Image fusion · attention map · long distance imaging · near infrared · pyramid feature selection.

1 Introduction

Image fusion is a key technology for information acquisition and processing which have many consumer applications such as video surveillance and autonomous cars. In recent years, sensor technology and image fusion have attracted much attention by researchers and industries. For image sensors, different imaging principles, wavelengths and environments lead to their own image characteristics. We usually choose an appropriate sensor depending on demand. Since a single sensor has certain limitation and cannot meet all needs, the image acquired by it are not able to perfectly reflect all the information in the scene. To solve this problem, image fusion is proposed to obtain more accurate and informative descriptions

This work was supported by the National Natural Science Foundation of China (No. 62111540272).

of the scene [13,21]. Since the characteristics of the images acquired by multiple sensors are different, it is required to analyze the characteristics of the multi-sensor images to complement each other and obtain images with clearer and higher-quality scene descriptions. Fusion of images from multiple sensors makes it easy to reliably observe and express the target information, which generates a fusion image with good contrast and fine details. Moreover, the fusion result is able to describe information better than a single image because image fusion eliminates redundant information in the source image. Image fusion has been widely used in many fields, including analysis and processing of remote sensing images [14], automatic recognition [24], computer vision [7], medical image processing [34], and security monitoring [22]. The fusion of color (RGB) and near infrared (NIR) images has become increasingly popular in the field of image fusion [3, 8–10, 15, 16, 33, 36]. Fig. 1 shows an RGB-NIR image pair captured in the same scene. NIR images are formed by an NIR sensor sensing NIR light. Different from visible light spectrum, the spectral range of NIR light is 750 nm to 2000 nm [30]. Since the imaging principles of NIR and RGB sensors are different, the obtained images are complementary. The NIR images distinguish the target from the background according to the radiation of the object in the NIR band, and are not easily affected by low light and bad weather. The RGB images are obtained by reflecting visible light. Its spectral information is much richer than NIR images. However, the anti-interference ability of the RGB image is poor, and it is difficult to obtain a clear image under severe weather conditions. Multispectral fusion of RGB and NIR images produces high quality images with fine textures and vivid colors by taking both advantages.

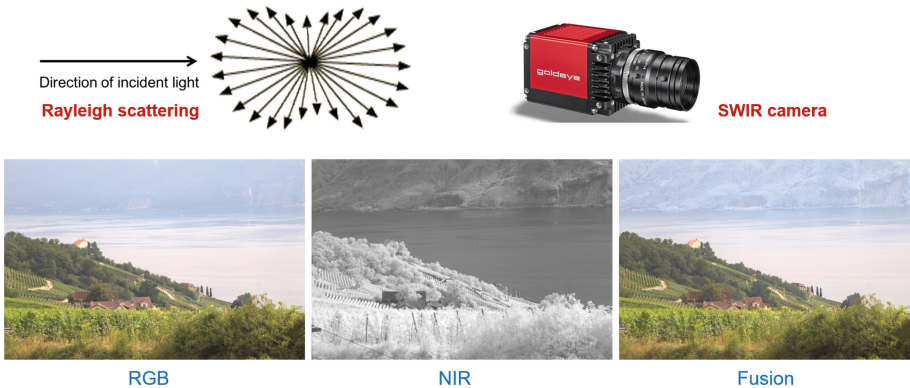


Fig. 1. Illustration of long distance imaging from a pair of RGB and NIR images. Top: Rayleigh scattering and short wave infrared (SWIR) camera. Bottom: RGB image, NIR image, and fusion result by LDINet. LDINet recovers hidden textures in the mountain by RGB-NIR image fusion. Rayleigh scattering is the phenomena of scattering of light particles and blurs distant areas.

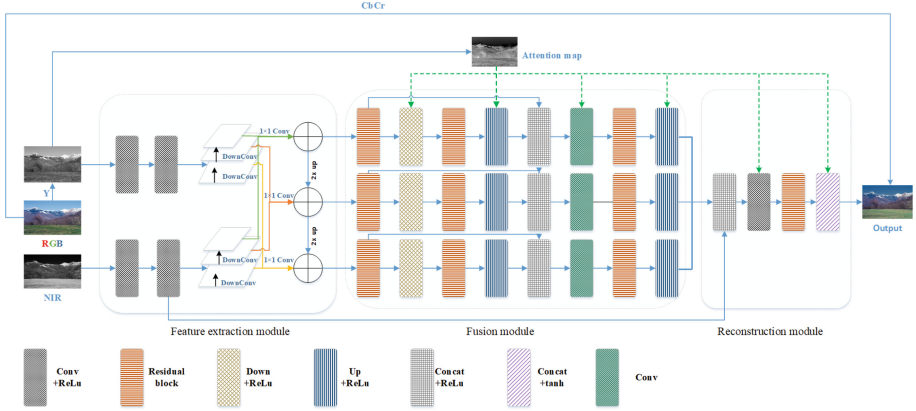


Fig. 2. Network architecture of LDINet for long distance imaging from a pair of RGB and NIR images. LDINet consists of three main modules: feature extraction, fusion and reconstruction.

In this paper, we propose long distance imaging through RGB and NIR image fusion, named LDINet. Rayleigh scattering makes it difficult for RGB cameras to capture long distance imaging such as sky and mountains due to the poor anti-interference ability as shown in Fig. 1. Moreover, areas exposed to direct sunlight are overexposed, while shadow areas are underexposed. Fog and clouds affect the quality of RGB images, but NIR images are robust to them. Since the overexposure and underexposure cause dynamic range allocation problems, we build a fusion network to guide contrast enhancement based on an attention map. Moreover, NIR images are not only robust to external environments such as light and atmospheric conditions, but also contain good details and contrast. However, traditional brightness fusion easily causes color shift, thus we introduce pyramid feature selection in the process of RGB and NIR fusion to get multi-scale features from RGB intensity channel and NIR image minimizing color distortion. As shown in Fig. 1, LDINet recovers hidden textures in the mountain by RGB-NIR image fusion. Compared with existing methods, the main contributions are as follows: 1) We use pyramid feature selection to transfer NIR details to the fusion while maintaining the original RGB tone. Pyramid feature selection extracts multi-scale information, which effectively retains the features of the shallow layer while improving the accuracy as the depth increases. 2) We generate an attention map from the RGB image to guide contrast enhancement during the fusion process. Thus, dark areas are more enhanced, while bright areas are less enhanced. 3) We synthesize a daylight image dataset for training based on smoothing operation, i.e. the input smoothed color images are generated by their original color images, i.e. ground truth.

2 Related Work

2.1 Image Fusion

In the 1970s, Meng et al. applied the image fusion technology to the military field [19]. Since then, many researchers have conducted research on image fusion under low light conditions. To meet the requirements of night vision applications, the sensitivity of the image sensor CCD/CMOS is continuously improved under low light conditions [11]. There have been many studies on using NIR images to restore RGB images containing strong noise under low light conditions [20, 27]. NIR images taken simultaneously in the same scene are very useful for RGB enhancement and restoration. NIR images carry brightness and spatial information without color, which contain good contrast and less noise even in low light condition. Schaul et al. [25] proposed a method of coloring directly on the NIR image to retain structures and details of NIR images. Unfortunately, due to different training datasets, the color of this fusion method is very different from the original RGB image. In 2015, Honda et al. [5] proposed to use NIR images for RGB image restoration under low-light conditions. Son et al. [28] proposed a contrast-preserving mapping model to produce an NIR image with a similar appearance in the luminance plane to the RGB image by preserving the contrast and details of the captured NIR image. In recent years, with the rapid development of artificial intelligence, it has become a trend to use deep learning for computer vision and image processing [4, 26]. Among them, convolutional neural networks (CNNs) in deep learning are popularly used [2, 6, 18]. Unlike the restoration of degraded images such as image denoising and super-resolution, the goal of image fusion is to fuse different types of images, and thus it is difficult to get the ground truth in image fusion. Vanmali and Gadrev [31] proposed a multi-resolution fusion method of RGB and NIR images based on Laplacian-Gaussian pyramid. They generated weight maps for image fusion using local entropy, local contrast and visibility. Jung et al. [9] proposed a fusion network of RGB and NIR images based on two stage CNNs, called FusionNet. They synthesized noisy RGB images for training data by adding noise in clean RGB images, and used the clean RGB images as ground truth. Jung et al. [10] proposed an unsupervised deep image fusion network with structure tensor representations, called DIF-Net. They designed an unsupervised loss function using structure tensor representation of the multi-channel image contrasts. Li et al. [15] proposed an encoder-decoder structure for the fusion of visible and infrared images based on CNNs, called DenseFuse. Then, they further proposed a CNN-based fusion framework that included encoder network, fusion strategy, and decoder network, called NestFuse [16]. The fusion strategy was based on spatial attention and channel attention models for the fusion of multiscale deep features. Recent image fusion methods such as DenseFuse [15] and NestFuse [16] mainly focus on the fusion of infrared and visible images. Although they can be used for the fusion of RGB and NIR images, infrared images do not contain details, thus they are much different from NIR images in texture and structure. Moreover, infrared images have a limit of considering atmospheric environments such as

Rayleigh scattering and Mie scattering that make long distance imaging of RGB camera difficult. Therefore, their fusion performance for RGB and NIR images is limited. So far, there are few methods designed for long distance imaging by the fusion of RGB and NIR images. In this work, we investigate a multispectral fusion network of RGB and NIR images based on pyramid feature selection and attention map to achieve long distance imaging.

2.2 Attention Map

For long distance imaging, it is required to consider the distance information in the fusion of RGB and NIR images. Therefore, we generate an attention map from the Y channel of RGB image. Generally, in an image, there is an obvious brightness difference between the near and distant regions, thus the distance information can be approximately estimated by the brightness of the image. The texture information of the near region is clear in the RGB image for long distance imaging, which should be retained in the fusion. Meanwhile, the texture information of the distant region is easily lost in the RGB image due to atmospheric environments such as Rayleigh scattering and Mie scattering. NIR images are robust to the atmospheric environments, which contain rich texture information especially in the distant region. Therefore, we use an attention map that approximately estimates the distance information for guiding image fusion. It has been reported that the dark channel prior (DCP) in RGB images roughly represents distance information by transmission [17]. Schaul et al. [12] proposed low light image enhancement based on unsupervised learning and bright channel priors (BCP). Similar to DCP, we generate the attention map from the Y channel of the RGB image. The attention map acts like a mask that assigns a small weight to the near region and a large weight to the distant region. The attention map contributes to estimating distance information in the fusion of RGB and NIR images.

2.3 Pyramid Feature Selection

Image pyramid is an effective and simple method, which is widely used in feature extraction [23,38], semantic segmentation [35] and image compression [29]. The image pyramid model is a group of images from the same original image arranged in a pyramid shape, whose resolution gradually decreases from bottom to top. It is obtained by gradual extraction and does not stop sampling until a certain termination condition is reached. The bottom of the pyramid is a high-resolution image, and the top is a low-resolution image. Therefore, comparing the images with the reduced resolution layer by layer and the pyramid, the higher the pyramid level, the smaller the image size and the lower the image resolution. Multi-scale feature extraction combines the extracted shallow and deep features. It directly uses all the extracted features, and the extracted shallow and deep features contain useless features, thus their fusion leads to redundancy. Therefore, a pyramid feature selection module is used to extract multi-scale information to obtain high-quality fusion images containing semantic information and textures.

3 Proposed Method

3.1 Network Architecture

Fig. 2 shows the entire network architecture of LDINet for RGB and NIR image fusion. We use the feature details of NIR image to enhance RGB image. By the fusion of RGB and NIR images, we achieve long distance imaging with good details and no color distortion. To consider the multispectral advantages of RGB and NIR images, we build a fusion network for hidden texture recovery based on pyramid feature selection and attention map. LDINet consists of three modules (feature extraction, fusion, and reconstruction) with attention map acquisition as follows:

Attention map acquisition: We convert RGB to YCbCr channel, and then invert the brightness of the Y channel to obtain the attention map. Since the Y channel contains the luma information of the image and the CbCr channels have the chroma information of the image, only the Y channel can be put into LDINet for training. It aims to not only accelerate the training speed of LDINet, but also ensure that the chroma information in the image is not lost. We use the attention map as an additional input during the fusion process to guide the contrast enhancement. As shown in Fig. 2, we multiply the attention map with the output of the corresponding layers to adjust contrast enhancement. Except the first and last convolution layers, the number of channels of all convolution layers is 64, and the size of convolution kernel is 3x3.

Feature extraction: The feature extraction module uses two convolution layers to extract shallow features from the Y channel and NIR image.

Fusion: The fusion module first generates three feature maps of large, medium, and small scales from the extracted features by pyramid feature selection. Then, each of the feature maps are concatenated and convoluted for fusion. Next, the fusion features are upsampled and added to fully fuse them at each scale. After then, the fusion features at different scales are convoluted 8 times to extract deep features. In the fusion module, the down sampling and up sampling operations are used to improve the processing speed, while skip connections are used to fuse shallow features with deep features preventing the gradient disappearance during training. We finally fuse the features by concatenating three scale features.

Reconstruction: In the reconstruction module, the fused features are convoluted and activated by tanh function to obtain the final fusion image.

3.2 Loss Function

To consider color, contrast, and details in fusion, we design the fusion loss function $L_{fusion-loss}$ based on color loss, contrast loss and detail loss as follows:

$$L_{fusion-loss} = L_{color-loss} + L_{contrast-loss} + L_{detail-loss} \quad (1)$$

where $L_{color-loss}$ is color loss, $L_{contrast-loss}$ is contrast loss, and $L_{detail-loss}$ is detail loss. To prevent color distortion, we use color loss. The reason of using

Gaussian blur in color loss is that it can remove high frequency details, making it easier to compare colors. Color loss is highly tolerant of small errors. Therefore, it can learn colors similar to the target image. The color loss is denoted as follows:

$$L_{color-loss}(X, Y) = L1(X_b, Y_b) \quad (2)$$

where b denotes Gaussian blur, X and Y denote the output and ground truth, respectively; and $L1$ represents L1 loss. We use Gaussian blur for the color loss to remove high frequency details, thus making the term tolerant to small errors. Therefore, this term learns to generate colors close to the target image. We aim to enhance contrast and reduce color distortion at the same time. In the optimal contrast-tone mapping (OCTM) [32], constraints are used to control the adverse side effects of contrast enhancement. Inspired by OCTM, we design contrast loss as follows:

$$L_{contrast-loss} = C + \lambda T = \sum (p_j s_j + \lambda w_j s_j) \quad (3)$$

where λ is Lagrange multiplier to regularize the relative importance of the two mutually conflicting fidelity metrics; p_j is the probability that a pixel in the fusion image has the input gray level j , s_j is the degree of change in the output intensity at level j in the fusion image. The contrast gain C depends only on the intensity distribution of the fusion image.

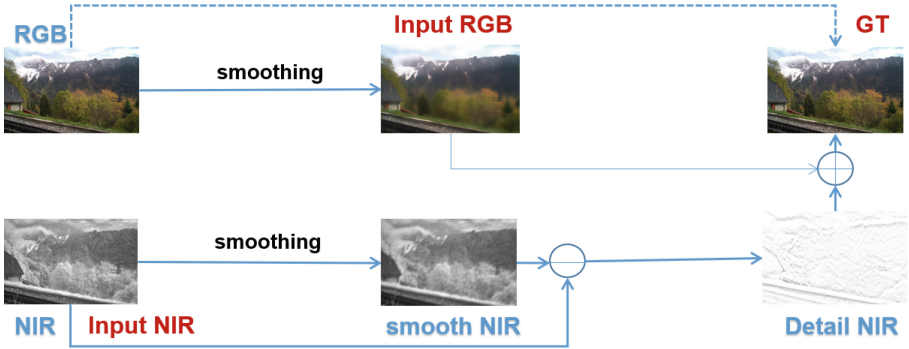


Fig. 3. Training data generation from RGB-NIR image pairs. The input smoothed color images (Input RGB) are generated by their original color images, i.e. ground truth (GT). To train LDINet, Input RGB and Input NIR are used as input, while GT is used as output.

The main goal of long distance imaging is to recover hidden textures in RGB image with the help of NIR image. Textural features are usually represented by the gradient of the image. To retain details from NIR images in fusion, we design the detail loss based on the image gradient by an inner product as follows:

$$L_{detail-loss} = L1(\nabla I_{fused}, Gra) - \mu < \nabla I_{fused}, \nabla I_{detail} > \quad (4)$$

where ∇I_{fused} is the gradient of the fusion image, Gra is the gradient of the ground truth, ∇I_{detail} is the gradient of the NIR image, $L1$ represents L1 loss, and μ is a weight to balance these two terms. We set μ to 0.1. In the detail loss, the first term makes the gradient of the fusion image close to that of the ground truth, while the second term makes the gradient of the fusion image close to that of NIR image for regularization. Since the fusion image needs to contain details for long distance imaging and NIR image contains them, the second term transfers details in NIR image to the fusion image.



Fig. 4. Sample image pairs of our training dataset. Left: RGB images. Middle: NIR images. Right: Ground truth.

3.3 Dataset Generation

For experiments, we construct training and testing sets from the RGB-NIR scene dataset [1]. The dataset contains 476 image pairs, some of which are not registered. Therefore, we select 99 registered image pairs for training and 18 registered image pairs for testing. For training, we crop 99 training image pairs to generate 13,500 pairs of 128x128 patches for training. Since upsampling and downsampling operations are used twice in LDINet, the height and width of the image need to be resized to an integer multiple of 4 before testing. The RGB-NIR scene dataset contains images captured in various scenes, which ensures the independence in the training and testing sets to a certain degree. At the same time, in the cropping process, we crop the image by a specific step size, which guarantees that the image patches in the training set are different from each other. In addition, during the training process we use random flipping for data augmentation to deal with the insufficient training data. As shown in Fig. 3, we first smooth the original color images as input, and fuse them with the NIR texture to restore the color image details. Then, we obtain NIR details by subtracting the smoothed



Fig. 5. Fusion results on RGB-NIR image pairs by LDINet. Left: Input RGB images. Middle: Input NIR images. Right: Fusion results.

NIR from the original NIR. We add the details of NIR images into the smoothed color images to generate the ground truth. In the image fusion process, we only want to fuse NIR details instead of brightness, thus we adjust the smoothing parameters. The smoothing parameter controls the degree of smoothing. The larger the value is, the smoother the image is. During the data generation process, the NIR smoothing parameter is adjusted to be smaller so that the weight of the detail NIR obtained by subtracting the smoothed NIR from the NIR. Thus, less NIR is added to the ground truth (GT). Usually, the smoothing parameters are in the range $[1e-3, 1e-1]$, where we choose $1e-2$. It can be seen from the figure that the input is smoothed RGB image, and the ground truth is generated by adding details of near infrared images to smooth color images. Based on our training dataset generation, we synthesize a set of synthetic images. First, we crop 90 RGB and NIR image pairs from 1024×680 to 128×128 , then randomly crop each image 150 times. Thus, we obtain 13,500 NIR, RGB (low light) and GT images, respectively. Fig. 4 shows some image pairs of our training dataset.

4 Experimental Results

For experiments, we use a PC with NVIDIA GeForce 1080ti 11GB GPU and Intel E5-2698 v4 @2.2GHz CPU, running Ubuntu 18.04 and Pytorch 1.6.0. The total epoch is 30, the batch size is set to 16, and the learning rate is set to $1e-3$. For tests, we select 18 pairs of RGB and NIR images from the RGB-NIR image dataset [1]. The RGB-NIR dataset contains 476 RGB and NIR image pairs: we use 99 image pairs for training data generation, and 18 image pairs for tests. Since it is difficult to generate the ground truth, we use the blind image quality evaluation (BIQE) for quantitative measurements [37]. Fig. 5 shows some fusion results on RGB-NIR image pairs by LDINet. As shown in the figure, LDINet

successfully recovers hidden textures lost in RGB images while keeping colors with the help of NIR images.



Fig. 6. Fusion results by different methods. Top: RGB image, NIR image, DenseFuse [15]. Bottom: DIF-Net [10], NestFuse [16], LDINet. LDINet generates a natural-looking fusion image with fine details while keeping the original color tone.

Visual Comparison: To verify the effectiveness of LDINet in fusion, we compare the fusion results by LDINet with state-of-the-art methods: DenseFuse [15], DIF-Net [10] and NestFuse [16]. The three methods are based on deep learning, and provide state-of-the-art fusion performance. Therefore, we select them for comparison to verify the performance of LDINet. Figs. 6 and 7 show fusion results by them on RGB-NIR image pairs. DenseFuse [15] and DIF-Net [10] can basically fuse the features of two images well, but they causes color shift after fusion. Their fusion results are much affected by the intensity of NIR images. Moreover, they contain halo artifacts along sharp edges between river and field. NestFuse [16] causes over-enhancement effects on the fusion results, thus making not natural-looking after fusion. Compared with them, LDINet successfully recovers hidden textures lost in the fusion results while keeping the original color tone. Thus, our fusion results are more natural-looking than the others.

Table 1. Average BIQE comparison among different methods. We obtain the BIQE scores in the table on all the test image pairs. The bold number represents the best performance (the smaller, the better).

Method	DenseFuse [15]	DIF-Net [10]	NestFuse [16]	Ours
BIQE [37]	21.9324	20.8448	21.3366	20.3189



Fig. 7. Fusion results by different methods. Top: RGB image, NIR image, DenseFuse [15]. Bottom: DIF-Net [10], NestFuse [16], LDINet. LDINet generates a natural-looking fusion image with fine details while keeping the original color tone.

Table 2. Average runtime comparison among different methods (Unit: sec/pair). The bold number represents the best performance (the smaller, the better). For tests, we use a PC with four NVIDIA GeForce 1080ti 11GB GPU and Intel E5-2698 v4 @2.2GHz CPU, running Ubuntu 18.04 and Pytorch 1.6.0.

Method	DenseFuse [15]	DIF-Net [10]	NestFuse [16]	Ours
Runtime	0.0020	0.0096	0.0863	0.0078

Quantitative Measurements: We provide average BIQE comparison among different methods in Table 1. We obtain the BIQE scores in the table on all the test image pairs. The bold number represents the best performance in BIQE, where the smaller the better. As shown in the table, LDINet outperforms the others in average BIQE score, which indicates that LDINet achieves the best visual quality after fusion. Moreover, we provide the average runtime of the fusion results among different methods in Table 2. The unit of runtime is sec/pair. DenseFuse [15] is the fastest in runtime, while LDINet ranks second in them.

Ablation Study: To see the effects of pyramid feature selection and color loss on the performance, we perform ablation experiments as follows. We obtain

Table 3. Ablation study on pyramid feature selection (pyramid) and color loss (color) in terms of BIQE [37]. We obtain the BIQE scores on the fusion results in Figs. 8 and 9. The lower the BIQE score is, the better the performance is.

Method	No pyramid & no color	No pyramid	No color	Ours
BIQE [37]	19.3720	19.6876	18.1184	16.8795

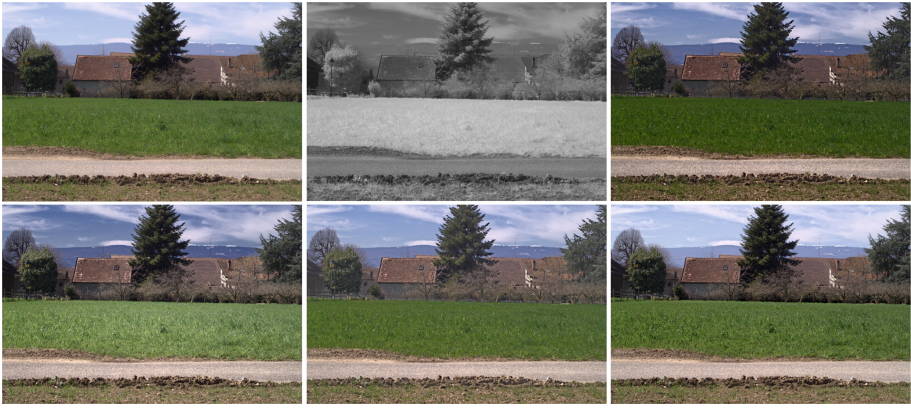


Fig. 8. Ablation study on pyramid feature selection and color loss. Top: RGB image, NIR image, fusion result without pyramid feature selection and color loss, Bottom: Fusion result without pyramid feature selection, fusion result without color loss, and fusion result by LDINet.



Fig. 9. Ablation study on pyramid feature selection and color loss. Top: RGB image, NIR image, fusion result without pyramid feature selection and color loss, Bottom: Fusion result without pyramid feature selection, fusion result without color loss, and fusion result by LDINet.

the fusion results without pyramid feature selection and color loss, those without pyramid feature selection, those without color loss in LDINet. Then, we compare them with the fusion results by LDINet. Figs. 8 and 9 show visual comparison of the fusion results by ablation experiments. The fusion results without pyramid feature selection and color loss look darker and less textures than those by LDINet. The fusion results without pyramid feature selection lose some details such as clouds and mountains. The fusion results without color loss contain color shift. However, LDINet generates natural-looking fusion images with fine details

while keeping the original color tone. Table 3 shows the BIQE scores on the fusion results in Figs. 8 and 9. The lower the BIQE score is, the higher the image quality is. The results indicate that both pyramid feature selection and color loss contribute to the fusion performance.

5 Conclusion

In this paper, we have proposed long distance imaging through RGB and NIR image fusion. We have constructed a fusion network of RGB and NIR images based on pyramid feature selection and attention map. We have used the attention map to guide contrast enhancement. Moreover, we have utilized pyramid feature selection to extract multi-scale information and generate a fusion image with fine details and good colors. Besides, we have generated training data that the input smoothed color images are generated by their original color images, i.e. ground truth. Experimental results demonstrate that LDINet generates natural-looking images from RGB-NIR image pairs and outperforms state-of-the-art methods based on deep learning in terms of visual quality and quantitative measurements.

When the input RGB and NIR images are not registered, LDINet may cause artifacts such as blur and halo along object boundaries. In our future work, we will investigate the registration issue in the fusion.

References

1. Brown, M., Süssstrunk, S.: Multi-spectral sift for scene category recognition. In: Proc. IEEE CVPR. pp. 177–184 (2011)
2. Guest, D., Cranmer, K., Whiteson, D.: Deep learning and its application to the physics. *Annu. Rev. Nucl. Part. Sci.* **68**, 161–181 (2018)
3. Han, Q., Jung, C., Zhou, K., Xu, Y.: Deep selective fusion of visible and near-infrared images using unsupervised u-net. *IEEE Transactions on Neural Networks and Learning Systems* (2022)
4. Hao, X., Zhang, G., Ma, S.: Deep learning. *International Journal of Semantic Computing* **10**(03), 417–439 (2016)
5. Honda, H., Timofte, R., Gool, L.: Make my day - high-fidelity color denoising with near-infrared. In: Proc. IEEE CVPRW. pp. 82–90 (2015)
6. Jia, F., Lei, Y., Guo, L., Lin, J., Xing, S.: A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. *Neurocomputing* **272**, 619–628 (2018)
7. Ju-Xia, L.I.: Difference feature recognition of computer vision image based on fusion. *Computer Simulation* (2015)
8. Jung, C., Han, Q., Zhou, K., Xu, Y.: Multispectral fusion of rgb and nir images using weighted least squares and convolution neural networks. *IEEE Open Journal of Signal Processing* **2**, 559–570 (2021)
9. Jung, C., Zhou, K., Feng, J.: Fusionnet: Multispectral fusion of rgb and nir images using two stage convolutional neural networks. *IEEE Access* **8**, 23912–23919 (2020)

10. Jung, H., Kim, Y., Jang, H., Ha, N., Sohn, K.: Unsupervised deep image fusion with structure tensor representations. *IEEE Trans. Image Process.* **29**, 3845–3858 (2020)
11. Laitinen, J., Ailisto, H.J.: Experimental evaluation of ccd and cmos cameras in low-light-level conditions. *Proc. SPIE* **3827**, 60–65 (1999)
12. Lee, H., Sohn, K., Min, D.: Unsupervised low-light image enhancement using bright channel prior. *IEEE Signal Process. Lett.* **27**, 251–255 (2020)
13. Li, B., Peng, H., Wang, J., Huang, X.: Multi-focus image fusion based on dynamic threshold neural p systems and surfacelet transform. *Knowl.-Based Syst.* **196**, 105794 (2020)
14. Li, D., Wang, Z., Li, Q.: Current progress on multisensor image fusion in remote sensing. *Proc. SPIE 4556, Data Mining and Applications* **5**(5), 1–6
15. Li, H., Wu, X.J.: Densefuse: A fusion approach to infrared and visible images. *IEEE Trans. Image Process.* **28**(5), 2614–2623 (2019)
16. Li, H., Wu, X.J., Durrani, T.: Nestfuse: An infrared and visible image fusion architecture based on nest connection and spatial/channel attention models. *IEEE Trans. Instrum. Meas.* **69**(12), 9645–9656 (2020)
17. Li, Y., Jung, C., Kim, J.: Single image depth estimation using edge extraction network and dark channel prior. *IEEE Access* **9**, 112454–112465 (2021). <https://doi.org/10.1109/ACCESS.2021.3100037>
18. Lin, C.J., Lin, C.H., Wang, S.H., Wu, C.H.: Multiple convolutional neural networks fusion using improved fuzzy integral for facial emotion recognition. *Appl. Sci.* **9**(13), 2593 (2019)
19. Meng, L., Liao, C., Wang, Z., Shen, Z.: Development and military applications of multi-source image fusion technology. *Aerospace Electronic Warfare* (2011)
20. Mikami, T., Sugimura, D., Hamamoto, T.: Capturing color and near-infrared images with different exposure times for image enhancement under extremely low-light scene. In: *Proc. IEEE ICIP*. pp. 669–673 (2015)
21. Pajares, G., De la Cruz, J.M.: A wavelet-based image fusion tutorial. *Pattern Recogn.* **37**(9), 1855–1872 (2004)
22. Petrusca, L., Cattin, P., De Luca, V., Preiswerk, F., Celicanin, Z., Auboiroux, V., Viallon, M., Arnold, P., Santini, F., Terraz, S.: Hybrid ultrasound/magnetic resonance simultaneous acquisition and image fusion for motion monitoring in the upper abdomen. *Invest. Radiol.* **48**(5), 333–340 (2013)
23. Rajaram, S., Rajendran, V., Abdullah, A.S., Suganya, R.: Prediction of heart diseases using hybrid feature selection and modified laplacian pyramid non-linear diffusion with soft computing methods. *Int. J. Biomed. Eng. Technol.* **25**(1), 30 (2017)
24. Samadzadegan, F., Schenk, T., Mahmoudi, F.T.: A multi-agent method for automatic building recognition based on the fusion of lidar range and intensity data. In: *Proceedings of the Joint Urban Remote Sensing Event* (2009)
25. Schaul, L., Fredembach, C., Süsstrunk, S.: Color image dehazing using the near-infrared. In: *Proc. IEEE ICIP*. pp. 1629–1632 (2010)
26. Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural Netw.* **61**, 85–117 (2015)
27. Socolinsky, D.A., Wolff, L.B.: *Face Recognition in Low-Light Environments Using Fusion of Thermal Infrared and Intensified Imagery*. Springer, London (2009)
28. Son, C.H., Zhang, X.P., Lee, K.: Near-infrared coloring via a contrast-preserving mapping model. In: *Proc. IEEE GlobalSIP*. pp. 678–681 (2015)
29. Song, X., Neuvo, Y.: Image compression using nonlinear pyramid vector quantization. *Multidimension. Syst. Signal Process.* **5**(2), 133–149 (1992)

30. Uchida, M., Ohmori, Y., Yoshino, K.: Electroluminescence from visible to near-infrared spectral range in buckminsterfullerene diode. *Jpn. J. Appl. Phys.* **30**(12B), L2104–L2106 (1991)
31. Vanmali, A.V., Gadre, V.M.: Visible and nir image fusion using weight-map-guided laplacian-gaussian pyramid for improving scene visibility. *Sādhanā* **42**, 1063–1082 (2017)
32. Wu, X.: A linear programming approach for optimal contrast-tone mapping. *IEEE Trans. Image Process.* **20**(5), 1262–1272 (2010)
33. Xu, H., Ma, J., Yuan, J., Le, Z., Liu, W.: Rfnet: Unsupervised network for mutually reinforcing multi-modal image registration and fusion. In: *Proc. IEEE CVPR*. pp. 19679–19688 (2022)
34. Yang, Y., Park, D.S., Huang, S., Rao, N.: Medical image fusion via an effective wavelet-based approach. *Eurasip Journal on Advances in Signal Processing* **2010**(579341), 1–13 (2010)
35. Yu, B., Yang, L., Chen, F.: Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **9**, 3252–3261 (2018)
36. Zhang, H., Mei, L., Jung, C.: Long range imaging using multispectral fusion of rgb and nir images. In: *Proc. IEEE ICASSP* (2023)
37. Zhang, L., Zhang, L., Bovik, A.C.: A feature-enriched completely blind image quality evaluator. *IEEE Trans. Image Process.* **24**(8), 2579–2591 (2015)
38. Zhao, G.P., Bo, Y.M.: Pyramid mean shift tracking algorithm based on adaptive feature selection. *Acta Photonica Sinica* (2011)



Efficient Long-Range Context Modeling for Motion Forecasting with State Space Models

Zhiwei Dong^(✉), Ran Ding, Jiaxiang Wang, and Wei Li

Huawei Riemann Lab, Xi'an, China
kivee@foxmail.com, liwei.levi@huawei.com

Abstract. Motion forecasting is a foundational task in autonomous driving, where accurate long-distance motion trajectories prediction of traffic participants depends heavily on modeling the long-range global context in traffic scenarios. Currently, the mainstream approach combines vectorized scene representation with the attention mechanism for context encoding. However, the inherent quadratic complexity of self-attention limits these attention-based methods' ability to fully encode long-range context due to the prohibitive computational costs. Consequently, they generally perform local attention as a trade-off between performance and efficiency. Inspired by the recent success of state space models with linear complexity in long sequence modeling, this paper introduces the Attention-SSM Block (ASB) to capture long-range contextual features for motion forecasting. The ASB starts by extracting local context using simple local attention, then sorts these tokens in a specific order and inputs them into a modified SSM, which considers relative position encodings between input tokens. We build an encoder based on ASB and combine it with a query-based decoder to form our motion forecasting model, MambaTraj. MambaTraj achieves excellent performance on the widely-used Argoverse2 benchmark with a small network parameter size and low inference latency. This confirms its effectiveness and efficiency in modeling long-range context for motion forecasting.

Keywords: Motion forecasting · State space models · Autonomous driving · Context modeling

1 Introduction

Motion forecasting of traffic participants (including vehicles, pedestrians, etc., hereafter referred to as agents) is a critical foundational task for autonomous driving. In this task, algorithms need to predict multiple possible future motion trajectories of these agents within a specific future period based on their historical movement information and environmental information such as high-definition (HD) maps. The challenge lies in the uncertainty of agents' intents, thus requiring comprehensive context feature encoding of complex traffic scenarios (including

agents’ movement features, map features, inter-agent interactions, and motion constraints imposed by the map) to support accurate motion forecasting.

Deep learning-based methods have now become the mainstream in motion forecasting. Inspired by the success of convolutional neural networks (CNN) in the field of computer vision, early deep learning-based motion forecasting methods rasterize traffic scenes as images in a top-down perspective view [2, 19, 23, 27] and then apply well-designed CNNs [16, 28] for context encoding. Although effective at the time, these methods struggle to model long-range contextual features due to the inherent receptive field mechanism of CNNs and accuracy loss of rasterized images. To overcome these shortcomings, current methods [31, 32, 39, 40] combine vectorized representation[8] with the attention mechanism [30]. The vectorized representation represents both agent movement and HD maps as vectors, which accurately describe the position of agents at each moment and the geometric information of HD map elements such as lane lines.

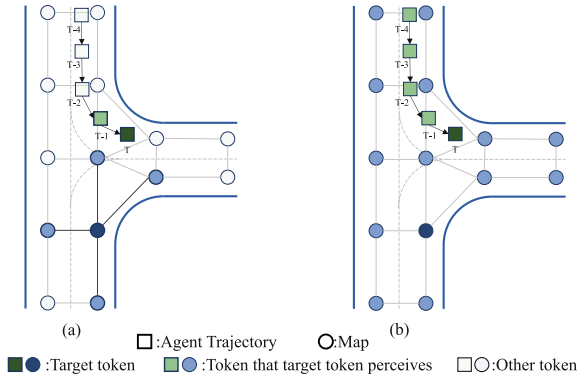


Fig. 1. Short-range and Long-range context modeling. (a) Due to the computational complexity constraints, each agent token and map token could only perceive a few tokens that are nearby in time or space. (b) By arranging tokens into a sequence and applying appropriate SSMs, each token can access information from all other tokens, and the entire process has linear computational complexity.

Although these vectorized representation-based methods have achieved state-of-the-art results, they have an inherent drawback. **The attention mechanism cannot efficiently encode the vectorized traffic scene’s long-range global context due to the limitation of huge computational costs.** This can be attributed to two main factors: firstly, the vectorized representation divides historical trajectories and HD map elements into many pieces; secondly, the attention mechanism has quadratic computational complexity. They always adopt a compromise solution by applying attention mechanisms in a local area or time span, as shown in Fig.1(a). Although tokens could perceive non-local information by applying local attention multiple times, it is not efficient to establish the long-range dependency between tokens. Hence, these methods have yet to

fully exploit the potential of vectorized representation. To facilitate efficient long-range context encoding for vectorized representation-based motion forecasting, we get inspired by the state space models (SSMs)[10–12, 15] in natural language processing (NLP) research. Recently, SSMs have gradually attracted attention due to their linear computational complexity and ability to handle long-range dependencies. The recent Mamba [9], in particular, allows SSM parameters to be functions of the input, enabling SSM to selectively propagate or forget information along the sequence length dimension depending on the current token. Mamba outperforms Transformers of the same size and performs on par with Transformers twice its size.

This paper proposes the **Attention-SSM Block** (ASB), which combines the powerful local feature extraction capability of local attention and the strong long-range dependency modeling capability of SSMs. Integrating attention mechanisms and SSMs poses two challenges. The first challenge arises from the fact that attention is permutation-invariant to the input, meaning the order of input tokens does not affect the result, while SSMs do not share this property. We propose to **sort the input tokens of SSMs for different context encodings** (agent movement features or map features) in a reasonable and stable manner, although there is no natural order for agents and HD map elements. The second challenge is that we want to combine SSMs with the relative position encoding approach [3, 17, 36], which has been proven to be more efficient than agent-centric [13, 29] and scene-centric [8, 21, 26] encoding methods in motion forecasting, while SSMs’ update process does not consider position encoding. Thus, we **introduce relative position encoding into SSMs**. To utilize ASB for the motion forecasting task, we introduce MambaTraj, an encoder-decoder structured motion forecasting model. MambaTraj constructs its encoder with ASB as the basic block and cooperates with a query-based decoder [1, 7].

Experiment results on Argoverse2 [33] benchmark demonstrate that MambaTraj achieves comparable performance with state-of-the-art methods but has fewer parameters and lower inference latency, indicating that SSMs can perform long-range context modeling effectively and efficiently.

2 Related Work

2.1 Scene Representation and Context Modeling

In motion forecasting, scene data obtained from offline ground truth or perception algorithms [6, 42] can be represented through two main methods: rasterization and vectorization. Rasterization converts maps, agent trajectories, and scene states into rasterized images suitable for processing by CNNs, but this method can suffer from information loss and a limited receptive field [2, 4]. Vectorization, which has gained popularity, represents scenes as sets of entities with semantic and geometric properties, enhancing the learning of entity relationships through graph convolutions and attention mechanisms [21, 34]. Vectorized representations are categorized into scene-centric and agent-centric approaches. Scene-centric methods use a single coordinate system for all agents, reducing

computational costs by representing the scene just once [8, 26]. Agent-centric methods normalize the coordinate system around each agent and represent the scene multiple times for accurate predictions [22, 25, 37].

Regarding context data processing, the historical trajectories of agents, map data, and traffic participant status are crucial. Methods like LaneGCN and BANet utilize graph neural networks (GNNs) to integrate multimodal data [21, 35], while SceneTransformer and Wayformer apply multi-axis attention to merge temporal and spatial information [25, 26]. However, the computational complexity of the self-attention mechanism in Transformers escalates with the length of the context, prompting state-of-the-art solutions to limit attention to localized areas or periods to manage computational demands [39, 40]. HiVT[40] normalizes the local context of each agent and explicitly merges the relative poses in local and global feature fusion to make the method viewpoint invariant. QCNet[39] uses a query-centric paradigm for scene encoding, which enables the reuse of past computations by learning representations independent of the global spacetime coordinate system.

2.2 State Space Models

State space models (SSMs) have been foundational in sequence modeling, evolving from hidden Markov models to sophisticated recurrent neural networks (RNNs). These models excel in handling sequences through recurrent updates of hidden states. The Structured State-Space Sequence (S4)[11] model represents a significant advance, optimizing computational efficiency through innovative reparameterization and attracting attention for its linear scaling ability with sequence length. Recent developments in SSMs have introduced linear-time attention variants such as H3[5] and Gated State Space[24], which enhance the efficiency and functional scope of these models. Mamba[9] builds upon these improvements by incorporating a data-dependent selection mechanism into S4, enabling more effective capture of long-range contexts as sequence lengths increase. Not only does Mamba demonstrate linear time efficiency, but it also exceeds the performance of traditional Transformers in various applications. Due to the strong long-range dependency modeling capability, Mamba has been successfully applied in vision tasks[14, 20, 41] recently. In this paper, we exploit its potential in motion forecasting.

3 Preliminaries

3.1 SSM for Sequence Modeling

SSM is inspired by a particular continuous system that maps a 1-D function or sequence $x(t) \in \mathbb{R} \rightarrow y(t) \in \mathbb{R}$ through a hidden state $h(t) = \mathbb{R}^N$. It could be formulated as:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t), \\ y(t) &= Ch(t), \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{N \times N}$, $B \in \mathbb{R}^{N \times 1}$ and $C \in \mathbb{R}^{1 \times N}$ are the system parameters. Mamba introduces a timescale parameter Δ and use zero-order hold (ZOH) to transform the system to a discrete version as follows:

$$\begin{aligned}\bar{A} &= \exp(\Delta A), \\ \bar{B} &= (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B, \\ h_t &= \bar{A}h_{t-1} + \bar{B}x_t, \\ y_t &= Ch_t.\end{aligned}\tag{2}$$

For a sequence of length L , the above discrete-time equations could be implemented through convolution as follows:

$$\begin{aligned}\bar{K} &= (C\bar{B}, C\bar{A}\bar{B}, \dots, C\bar{A}^{L-1}\bar{B}), \\ y &= x * \bar{K}.\end{aligned}\tag{3}$$

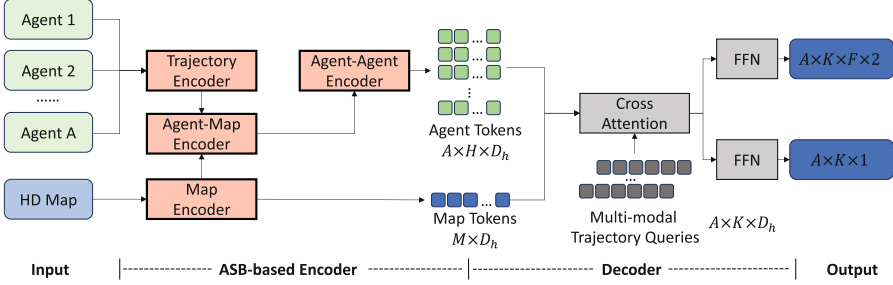


Fig. 2. Architecture of MambaTraj. The inputs are agents’ historical movement information and the HD map. The outputs are predicted trajectories and confidence scores. The encoder is constructed with four sub-encoders based on ASB and produces agent and map tokens. The decoder uses a set of learnable multi-modal trajectory queries to extract features from agent, and map tokens and decodes predicted trajectories and confidence scores through two FFNs.

3.2 Problem Formulation

The task of motion forecasting involves predicting the future motion trajectories of agents based on the input of their historical trajectory information and the HD map of the environment in which these agents are located. The past H frames of trajectories of A input agents (vehicle, pedestrian, etc.) are typically represented as $A \times H \times D_a$, where D_a includes 2D position as well as other attributes such as the agent’s category. The corresponding K possible future trajectories of F frames to be predicted are $A \times K \times F \times 2$. As for the representation of the map, we employ the widely-used vectorized representation [8, 39, 40]. It represents

map information as a set of lane line segments shaped $M \times D_m$ (where D_m includes the start and end point coordinates of each lane segment, as well as their attributes, such as turns, intersections, bus lanes, etc.), with each lane segment not exceeding 5 meters in Argoverse2. It should be noted that A , M , H , F , D_a and D_m are not hyperparameters which need to be tuned. A and M vary in different scene samples. H , F , D_a and D_m are provided by the benchmarks.

4 Methodology

4.1 Network Architecture

As illustrated in Figure 2, MambaTraj has a typical encoder-decoder structure. The encoder is composed of four sub-encoders: trajectory encoder, map encoder, agent-map encoder, and agent-agent encoder. All sub-encoders consist of multiple stacked Attention-SSM blocks. The trajectory encoder encodes the historical trajectory information of all agents as $A \times H \times D_h$, while the map encoder encodes the vectorized map information as $M \times D_h$. Subsequently, the agent-map encoder uses the map features obtained from the map encoder to enhance the agent features obtained from the trajectory encoder. The agent-agent encoder finally models the relationships between agents to further extract their interactive features. In the decoder, there is a set of shared learnable multi-modal queries for all agents, shaped $K \times D_h$. To decode the trajectories of all the A agents in parallel, we replicate queries to $A \times K \times D_h$. These queries extract the features necessary for predicting future motion trajectories through multiple cross-attention layers from the feature tokens of agents and maps generated by the encoder, thereby producing a $A \times K \times D_h$ feature vector for agents. This feature vector is then processed through two separate feed-forward networks (FFNs) to yield the predicted K possible future trajectories $A \times K \times F \times 2$ and their corresponding confidence scores $A \times K \times 1$ for all agents.

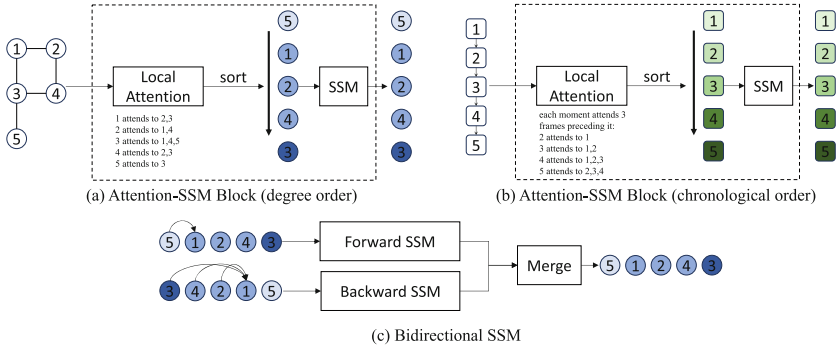


Fig. 3. Attention-SSM block and the bidirectional SSM.

4.2 Attention-SSM Block

To leverage the long-range context modeling capabilities of the SSM, we combine it with local attention and introduce the Attention-SSM Block (ASB). As shown in Fig.3(a) and Fig.3(b), the input to an ASB is a set of tokens (agent features, lane segment features, or both of them). Local attention is applied to each token and its nearby tokens in space or time to extract local features of input tokens. If there are N tokens and, on average, each token has k neighbors (where k is much smaller than N), then the complexity of this step is $O(kN)$. Subsequently, the tokens are sorted appropriately to form a token sequence as the input into the SSM for extracting long-range contextual features, with this step having a complexity of $O(N)$. If attention were used for complete long-range context modeling, each token would attend to all other tokens, leading to a computational complexity of $O(N^2)$, which is highly impractical when N is large in those complex traffic scenes. The analysis of computational complexity demonstrates that our Attention-SSM block, compared to a purely attention-based structure, can perform complete long-range context modeling much more efficiently.

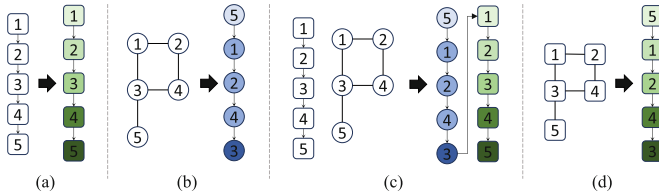


Fig. 4. Different sorting strategies. (a) Chronological order for the historical movement tokens of one agent; (b) Sort map tokens by their degrees; (c) Sort trajectory tokens of one agent chronologically and sort map tokens by degrees, then concatenate trajectory after map; (d) Sort different agents' tokens at the same moment by their degrees.

4.3 Sorting Strategy of Tokens in ASB-based Encoders

The four sub-encoders of MambaTraj are composed of multiple ASBs stacked together. Due to the different semantics of the tokens they process, the internal computation processes (especially the sorting of tokens) within their ASBs vary. Below, we provide a detailed description to the computation processes and token sorting strategies within the ASBs of these sub-encoders.

Trajectory Encoder The input to the trajectory encoder consists of multiple agents' historical movement information, represented as $A \times H \times D_a$. The trajectory encoder encodes the trajectory features for A agent independently along the temporal dimension. In the local attention phase, each moment attends only to a small period h preceding it to extract local temporal motion features. Subsequently, we arrange the feature tokens of different moments chronologically for

each agent (Fig.4(a)) and input them into the SSM. Since the tokens are placed in chronological order, each token can perceive all moments before it through the SSM, allowing for the extraction of long-range temporal motion features. The final agent features obtained are $A \times H \times D_h$.

Map Encoder The input of the map encoder consists of features $M \times D_m$ of all lane segments in the agent’s surrounding environment. In local attention, each lane segment only attends to other segments within a distance range r , which is set to 50 meters in our method. In the sort step, we sort the lane segment tokens based on their degree from lowest to highest (Fig.4(b)). *The intuition is that tokens with a higher degree are often located at traffic hubs, such as the central area of intersections, and they should be placed towards the end of the sequence, allowing them to capture the features of the majority of other tokens.* For tokens with the same degree, we perform random sorting during the training process to make the SSM more robust to the ordering noise of tokens with the same degree. During inference, we apply multiple random orderings to these same-degree tokens to form multiple sequences and then merge the inference results of these sequences. However, since we focus on modeling the long-range relationship between all map tokens, we need to use a bidirectional SSM [41] as shown in Fig.3(c). It is important to note that using a bidirectional SSM does not contradict sorting map tokens by degree because the bidirectional SSM also requires a sequentially stable input sequence. The final map features obtained are $M \times D_h$.

Agent-Map Encoder The agent-map encoder is designed to enhance agent features by enabling agents to acquire information about their surrounding environment through map tokens. Its inputs are agent tokens $A \times H \times D_h$ and map tokens $M \times D_h$. In local attention, each agent token attends to map tokens within the distance range r . As we want agents to perceive complete map information in the SSM, we repeat map tokens to $A \times M \times D_h$ and concatenate each agent’s H tokens after map tokens to get $A \times (M + H) \times D_h$. Within the map tokens and agent tokens, they are sorted according to degree and chronological order, respectively (Fig.4(c)). After processing the A sequences through the SSM, we take the last H tokens from each sequence to recover $A \times H \times D$ agent tokens.

Agent-Agent Encoder The input to the agent-agent encoder is agent tokens $H \times A \times D_h$, with the aim to model the interactions between A agents at H different moments. For local attention, an agent at moment t only attends to other agents within a certain distance range r at the same moment instead of considering all other agents and other moments. Subsequently, for H sequences of A agent tokens each, they are sorted by degree (Fig.4(d)). Then, they enter a bidirectional SSM, allowing each agent to efficiently extract features from all other agents at the same moment regardless of the distance.

4.4 Relative Position Encoding in SSMs

To encode agent and map features efficiently in a rotation and translation invariant manner, we follow the scene representation method of [39, 40]. We normalize each moment’s motion vector of the agent’s historical trajectory with the position at that moment as the origin and the heading direction as the positive direction of the x-axis. Similarly, we normalize each lane line segment vector with its start point as the origin and its direction as the positive direction of the x-axis. Thus, when using attention and SSMs to encode agent and map features, it is necessary to know the relative positions between different moments of an agent, between different lane segments, between agents and lane segments, and between different agents. Otherwise, they cannot model their surrounding environment correctly.

In methods using attention for context encoding [3, 17, 36, 39, 40], the information of keys and values relative to the query is incorporated into the attention computation process as follows:

$$\begin{aligned} q_i &= W^Q h_i, \\ k_{ij} &= W^K h_j + W^{r \rightarrow k} r_{ij}, \\ v_{ij} &= W^V h_j + W^{r \rightarrow v} r_{ij}, \end{aligned} \quad (4)$$

where r_{ij} means token j ’s geometric information relative to token i (relative position, relative orientation, etc.), and h represents agent or map feature tokens from the encoder. The function of the above formulas could be viewed as converting the feature of token j in its local coordinate system into the local coordinate system of token i through adding the relative geometric information in a latent high-dimension space, enabling token i to extract the features of its surrounding context in its own local coordinate system.

We could directly use the SSM as Eq.(2) to process the feature token sequence of agents or lane segments in their local coordinate systems. x_t is the feature of token t in its local coordinate system, which first fuses with the hidden state of token $t - 1$ to obtain token t ’s local coordinate system hidden state and then undergoes a feature transformation to obtain the output token y_t that perceives token t and all its previous tokens. However, a conflict arises in this process: h_{t-1} is in the local coordinate system of token $t - 1$, while x_t is in the local coordinate system of token t , making direct fusion between them unreasonable. Therefore, we propose to correct this process using the relative geometric information of token $t - 1$ to token t , with the formula:

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + W^{r \rightarrow h} r_{t,t-1} + \bar{B}x_t, \\ y_t &= Ch_t. \end{aligned} \quad (5)$$

This formula can be intuitively understood as first transforming the hidden state of token $t - 1$ to the local coordinate system of token t through adding $W^{r \rightarrow h} r_{t,t-1}$, and then fusing it with the feature of token t . To implement the above equations through convolution, we could precompute $W^{r \rightarrow h} r_{t,t-1}$ at all

time steps in the token sequence and use them as part of the input, denoted as r . The convolution version of Eq.(5) is as follows:

$$\begin{aligned}\bar{K} &= (C\bar{B}, C\bar{A}\bar{B}, \dots, C\bar{A}^{L-1}\bar{B}), \\ \bar{M} &= (C, C\bar{A}, \dots, C\bar{A}^{L-1}), \\ y &= x * \bar{K} + r * \bar{M}.\end{aligned}\tag{6}$$

4.5 Decoder and Loss Functions

In the decoder, there is a set of learnable $K \times D_h$ query embeddings shared by all agents. For each agent, the query embedding interacts with $A \times H \times D_h$ agent tokens and $M \times D_h$ map tokens through cross-attention layers to extract features for predicting future trajectories. Subsequently, using two FFNs to decode K possible future trajectories $A \times K \times F \times 2$ in the local coordinate system of each agent at the moment H and the confidence scores $A \times K \times 1$.

Similar to [40], we use the Laplace Negative Log Likelihood loss for the trajectory regression, requiring the output of the FFN decoding the trajectory to be $A \times K \times F \times 4$, which includes K sets of 2D coordinates $\mu_i^t \in \mathbb{R}^2$ for F moments and their corresponding uncertainty $b_i^t \in \mathbb{R}^2$. The regression loss function for the trajectory is shown as follows:

$$L_{reg} = -\frac{1}{AF} \sum_{i=1}^A \sum_{t=H+1}^{H+F} \log P(pos_i^t | \hat{\mu}_i^t, \hat{b}_i^t),\tag{7}$$

where $\hat{\mu}_i^t, \hat{b}_i^t$ represent the coordinates and uncertainty at moment t of the best predicted trajectory of the i -th agent, and $P(\cdot)$ is the probability density function of Laplace distribution. For the learning of confidence score, we use the cross-entropy loss as confidence loss L_{conf} . To get the target confidence score, we use *Softmax* to normalize the negative values of the errors between K predicted trajectories and the ground truth. Finally, we train MambaTraj using:

$$L_{total} = L_{reg} + L_{conf}.\tag{8}$$

5 Experiments

5.1 Experiment Settings

Dataset Argoverse2 [33] is the latest large-scale motion forecasting dataset collected in six cities, such as Miami and Pittsburgh, totaling 250,000 scenarios. It is divided into 200,000 scenarios for training, 25,000 for validation, and 25,000 for testing. The ground truth of the test split is not released, so model prediction results of the test split must be submitted to the official server to obtain metrics. In each scenario, it provides 5-second historical trajectories of various traffic participants (vehicles, pedestrians, bicycles, buses, etc.) sampled at 10Hz ($H = 50$) and the HD map of the current scene, encompassing details of lane lines, sidewalks, and other traffic elements. The model is required to predict a 6-second future trajectory of each agent, that is $F = 60$.

Metrics We follow the official evaluation metrics of Argoverse2, which include $\min FDE_k$, $\min ADE_k$, MR_k , and $brier - \min FDE_k$. These metrics are calculated based on the best one out of k predicted trajectories, where the official metrics commonly use $k = 1$ and $k = 6$. FDE refers to the L2 distance between the predicted trajectory and the ground truth trajectory at the last frame, while ADE is the average L2 error across all F predicted time steps. MR (Miss Rate) refers to the proportion of predicted trajectories with an FDE larger than 2 meters. Brier-minFDE is calculated as $(1 - p)^2 \times \min FDE$, where p is the confidence score of the best-predicted trajectory. All these metrics are designed such that lower values indicate better performance.

Table 1. Comparison of different methods on the test split of Argoverse2 dataset. \star means using model ensemble techniques.

Method	$\min FDE_k \downarrow$		$\min ADE_k \downarrow$		$MR_k \downarrow$		$b-FDE_6 \downarrow$
	k=1	k=6	k=1	k=6	k=1	k=6	
HDGT[17]	5.37	1.60	2.08	0.84	0.66	0.21	2.24
HPTR[36]	4.61	1.43	-	0.73	-	0.19	2.03
GoRela[3]	4.62	1.48	1.82	0.76	0.66	0.22	2.01
GANet[31]	4.47	1.35	1.77	0.71	0.59	0.17	1.96
QCNet[39]	4.3	1.29	1.69	0.65	0.59	0.16	1.91
ProphNet[32]	4.74	1.33	1.8	0.68	0.61	0.18	1.88
SmartRefine[38]	4.17	1.23	1.65	0.63	0.58	0.15	1.86
MambaTraj	4.1	1.25	1.6	0.63	0.57	0.16	1.89
QCNet \star	3.96	1.19	1.56	0.62	0.55	0.14	1.78
SEPT \star [18]	3.7	1.15	1.49	0.61	0.54	0.14	1.74
MambaTraj \star	3.74	1.17	1.5	0.61	0.54	0.13	1.76

Table 2. Comparison of different methods' efficiency on Argoverse2 dataset.

Method	codes available	Param \downarrow	Latency(ms) \downarrow
HPTR[36]	✓	10.3M	335
QCNet[39]	✓	7.7M	182
ProphNet[32]	✗	-	28
SmartRefine[38]	✓	8.0M	207
MambaTraj	Ours	2.9M	30

5.2 Comparison with State of the Art

Table 1 shows that MambaTraj achieves comparable performance with state-of-the-art published methods on the Argoverse2 test split, both with and without the ensemble technique. QCNet utilizes query-based decoder and hierarchical encoder[40] which combines local and global attention. Compared to QCNet, MambaTraj’s advantage suggests that SSMs could better model long-range context than global attention. When applying the ensemble technique, SEPT[18] has slightly better results than MambaTraj, because it uses self-supervised pretraining task on additional data, including the validation and test splits of Argoverse2, to get a stronger encoder. We also compare MambaTraj with other methods in terms of parameter size and average inference latency in Table 2. Except ProphNet, other models’ parameter counts are obtained from their official open-source codes and we test their inference latency on a single V100 GPU. We can only get the inference latency of ProphNet from its published paper[32] and cannot get its parameter size as the authors did not release codes. [18] neither reports parameter size nor speed in the paper, nor does it release code, so we cannot compare efficiency with SEPT. Table 1 and Table 2 show that MambaTraj could achieve better performance with fewer parameters and have low inference latency as well as the ProphNet, which sacrifices performance for computing efficiency. These results demonstrate that our method can effectively and efficiently model long-range contextual features, improving the accuracy of trajectory predictions.

5.3 Ablation Study

To validate the effectiveness of ASB, we replace the ASB in all sub-encoders with blocks composed solely of local attention (2nd row) and SSMs (3rd row), maintaining a similar parameter size. The experiment results are displayed in Table 3, showing our standard model (1st row), with all sub-encoders built upon ASB, has the best performance. Replacing attention with SSM alone even leads to worse performance, suggesting that SSMs may not be suitable for extracting local context and need to be combined with attention for better results.

In the 4th to 7th rows, **ASB-G** represents that we replace the SSM in ASB with the global attention. The results demonstrate that, compared to the global attention mechanism, SSM is more efficient for long-range context encoding.

In the 8th to 12th rows, **ASB-S** represents that we remove the sorting step in ASB. The results indicate that the correct ordering of tokens input into the SSM is crucial for the ASB to function correctly, and the sorting step incurs only a very small time cost, primarily due to the computation of node degrees.

In the 13th to 17th rows, **ASB-R** represents that we remove the relative position encoding injection in SSM. The results suggest that when combining relative position encoding approaches with SSM, it is necessary to inject the relative information between tokens into the SSM’s update process through Eq.(5). Otherwise, the ASB cannot yield obvious improvements compared to pure local attention, although it has lower inference latency.

Table 3. Ablation study of ASB in different sub-encoder components of MambaTraj on the validation split of Argoverse2.

Row ID	Traj.	Map	A-M	A-A	minFDE ₆	minADE ₆	Latency(ms)
1	ASB	ASB	ASB	ASB	1.26	0.71	31
2	Loc-Att	Loc-Att	Loc-Att	Loc-Att	1.38	0.77	40
3	SSM	SSM	SSM	SSM	1.54	0.91	23
4	ASB- <i>G</i>	ASB	ASB	ASB	1.30	0.71	64
5	ASB	ASB- <i>G</i>	ASB	ASB	1.32	0.73	132
6	ASB	ASB	ASB- <i>G</i>	ASB	1.28	0.72	86
7	ASB	ASB	ASB	ASB- <i>G</i>	1.29	0.72	45
8	ASB- <i>S</i>	ASB	ASB	ASB	1.53	0.80	31
9	ASB	ASB- <i>S</i>	ASB	ASB	1.39	0.76	28
10	ASB	ASB	ASB- <i>S</i>	ASB	1.48	0.79	29
11	ASB	ASB	ASB	ASB- <i>S</i>	1.31	0.73	30
12	ASB- <i>S</i>	ASB- <i>S</i>	ASB- <i>S</i>	ASB- <i>S</i>	1.72	0.94	27
13	ASB- <i>R</i>	ASB	ASB	ASB	1.30	0.72	31
14	ASB	ASB- <i>R</i>	ASB	ASB	1.31	0.73	31
15	ASB	ASB	ASB- <i>R</i>	ASB	1.27	0.72	31
16	ASB	ASB	ASB	ASB- <i>R</i>	1.29	0.72	31
17	ASB- <i>R</i>	ASB- <i>R</i>	ASB- <i>R</i>	ASB- <i>R</i>	1.37	0.76	31

5.4 Qualitative Results

To better understand the advantage of our approach, we visualize some typical prediction results of QCNet and MambaTraj in complex traffic scenarios in Fig.5. Columns one to three are all intersection scenarios, where it can be observed that MambaTraj provides more comprehensive predicted trajectories (blue arrows) than QCNet. This is attributed to the long-range context modeling capability of SSMs, allowing MambaTraj to more accurately and comprehensively perceive traffic situations at intersections. The fourth column presents a lane-changing overtaking scenario. QCNet’s predicted trajectories are quite short, whereas MambaTraj can accurately predict a longer trajectory following the overtaking maneuver. These qualitative results demonstrate that the long-range context encoding of the ASB module takes into account both the map information and motion information of other vehicles.

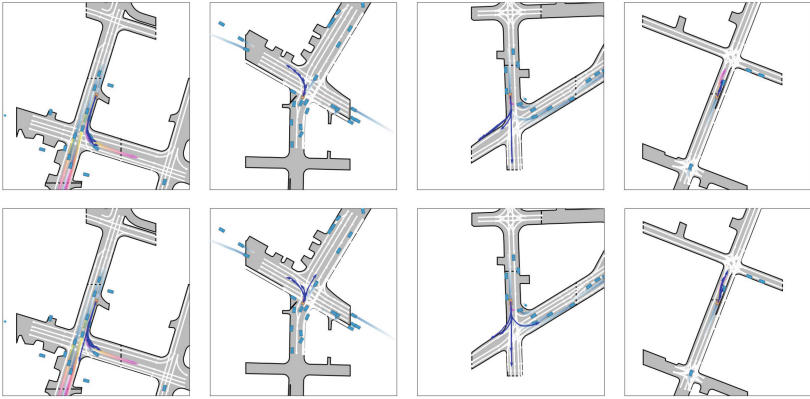


Fig. 5. Qualitative results of QCNet (upper row) and MambaTraj (lower row) on the validation split of Argoverse2.

6 Conclusions

This paper introduces MambaTraj, which integrates the state space models with the motion forecasting task. The core of MambaTraj is the novel Attention-SSM Block (ASB), which combines the local features extraction ability of the attention mechanism with the efficient long-range dependency modeling capability of SSMs. ASB employs appropriate token sorting strategies and relative position encoding to adapt the SSMs to the vectorized scene representation. Consequently, MambaTraj can efficiently and comprehensively perform both local and global context encoding for complex vectorized traffic scenarios. The performance metrics and qualitative results on the latest large-scale motion forecasting benchmark, Argoverse2, indicate the effectiveness of our method.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
2. Chai, Y., Sapp, B., Bansal, M., Anguelov, D.: Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. arXiv preprint [arXiv:1910.05449](https://arxiv.org/abs/1910.05449) (2019)
3. Cui, A., Casas, S., Wong, K., Suo, S., Urtasun, R.: Gorela: Go relative for viewpoint-invariant motion forecasting. In: 2023 IEEE International Conference on Robotics and Automation
4. Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: 2019 international conference on robotics and automation
5. Dao, T., Fu, D.Y., Saab, K.K., Thomas, A.W., Rudra, A., Ré, C.: Hungry hungry hippos: Towards language modeling with state space models. In: Proceedings of the 11th International Conference on Learning Representations (ICLR) (2023)

6. Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., Qian, C.: Centripetalnet: Pursuing high-quality keypoint pairs for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2020)
7. Dong, Z., Zhu, X., Cao, X., Ding, R., Li, W., Zhou, C., Wang, Y., Liu, Q.: Bezier-former: A unified architecture for 2d and 3d lane detection. In: IEEE International Conference on Multimedia and Expo (2024)
8. Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., Schmid, C.: Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In: CVPR (2020)
9. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint [arXiv:2312.00752](https://arxiv.org/abs/2312.00752) (2023)
10. Gu, A., Goel, K., Gupta, A., Ré, C.: On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems* (2022)
11. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint [arXiv:2111.00396](https://arxiv.org/abs/2111.00396) (2021)
12. Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems* (2021)
13. Gu, J., Sun, C., Zhao, H.: Densentn: End-to-end trajectory prediction from dense goal sets. In: ICCV (2021)
14. Guo, H., Li, J., Dai, T., Ouyang, Z., Ren, X., Xia, S.T.: Mambair: A simple baseline for image restoration with state-space model. arXiv preprint [arXiv:2402.15648](https://arxiv.org/abs/2402.15648) (2024)
15. Gupta, A., Gu, A., Berant, J.: Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems* (2022)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: ICCV (2016)
17. Jia, X., Wu, P., Chen, L., Liu, Y., Li, H., Yan, J.: Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *IEEE transactions on pattern analysis and machine intelligence* (2023)
18. Lan, Z., Jiang, Y., Mu, Y., Chen, C., Li, S.E.: Sept: Towards efficient scene representation learning for motion prediction. In: International Conference on Learning Representations (ICLR) (2024)
19. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: ICCV (2017)
20. Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y.: Videomamba: State space model for efficient video understanding. arXiv preprint [arXiv:2403.06977](https://arxiv.org/abs/2403.06977) (2024)
21. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: ECCV (2020)
22. Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B.: Multimodal motion prediction with stacked transformers. In: CVPR (2021)
23. Marchetti, F., Becattini, F., Seidenari, L., Bimbo, A.D.: Mantra: Memory augmented networks for multiple trajectory prediction. In: CVPR (2020)
24. Mehta, H., Gupta, A., Cutkosky, A., Neyshabur, B.: Long range language modeling via gated state spaces. In: International Conference on Learning Representations (ICLR) (2023)

25. Nayakanti, N., Al-Rfou, R., Zhou, A., Goel, K., Refaat, K.S., Sapp, B.: Wayformer: Motion forecasting via simple & efficient attention networks. In: 2023 IEEE International Conference on Robotics and Automation
26. Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al.: Scene transformer: A unified architecture for predicting multiple agent trajectories. arXiv preprint [arXiv:2106.08417](https://arxiv.org/abs/2106.08417) (2021)
27. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In: ECCV (2020)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
29. Varadarajan, B., Hefny, A., Srivastava, A., Refaat, K.S., Nayakanti, N., Cornman, A., Chen, K., Douillard, B., Lam, C.P., Anguelov, D., et al.: Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction. In: 2022 International Conference on Robotics and Automation
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* (2017)
31. Wang, M., Zhu, X., Yu, C., Li, W., Ma, Y., Jin, R., Ren, X., Ren, D., Wang, M., Yang, W.: Ganet: Goal area network for motion forecasting. In: 2023 IEEE International Conference on Robotics and Automation
32. Wang, X., Su, T., Da, F., Yang, X.: Prophet: Efficient agent-centric motion forecasting with anchor-informed proposals. In: CVPR (2023)
33. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint [arXiv:2301.00493](https://arxiv.org/abs/2301.00493) (2023)
34. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: ECCV (2020)
35. Zhang, C., Sun, H., Chen, C., Guo, Y.: Banet: Motion forecasting with boundary aware network. arXiv preprint [arXiv:2206.07934](https://arxiv.org/abs/2206.07934) (2022)
36. Zhang, Z., Liniger, A., Sakaridis, C., Yu, F., Gool, L.V.: Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding. *Advances in Neural Information Processing Systems* (2024)
37. Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., Schmid, C., et al.: Tnt: Target-driven trajectory prediction. In: Conference on Robot Learning (2021)
38. Zhou, Y., Shao, H., Wang, L., Waslander, S.L., Li, H., Liu, Y.: Smartrefine: A scenario-adaptive refinement framework for efficient motion prediction. In: CVPR (2024)
39. Zhou, Z., Wang, J., Li, Y.H., Huang, Y.K.: Query-centric trajectory prediction. In: CVPR (2023)
40. Zhou, Z., Ye, L., Wang, J., Wu, K., Lu, K.: Hivt: Hierarchical vector transformer for multi-agent motion prediction. In: CVPR (2022)
41. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint [arXiv:2401.09417](https://arxiv.org/abs/2401.09417) (2024)
42. Zhu, X., Cao, X., Dong, Z., Zhou, C., Liu, Q., Li, W., Wang, Y.: Nemo: Neural map growing system for spatiotemporal fusion in bird's-eye-view and bdd-map benchmark. arXiv preprint [arXiv:2306.04540](https://arxiv.org/abs/2306.04540) (2023)



DWT-SALF: Subband Adaptive Neural Network Based In-Loop Filter for VVC Using Cyclic DWT

Yunfeng Liu and Cheolkon Jung^(✉)

School of Electronic Engineering, Xidian University, Xian 710071, China
zhengzk@xidian.edu.cn

Abstract. In this paper, we propose a subband adaptive neural network based in-loop filter in VVC using cyclic discrete wavelet transform (DWT), named DWT-SALF. DWT-SALF takes advantages of subband adaptive learning based on DWT in the neural network-based in-loop filter (NNLF). Compared to the convolutional neural network (CNN), transformer is effective in capturing low-frequency features but has limited ability of constructing high-frequency representations. Thus, DWT-SALF uses transformer to handle the low-frequency subband, while utilizing CNN to treat the high-frequency subbands. We further enhance the high-frequency subbands with the guidance of the processed low-frequency subband. To increase the network depth and receptive field without increasing parameters, we adopt cyclic DWT that is cyclically used twice in the basic block and its affiliated branches of high and low frequency. Experimental results show that DWT-SALF achieves significant BD-rate gains of $\{-8.10\%$ (Y), -21.19% (U), -22.28% (V) $\}$ over the VTM-11.0_NNVC-2.0 anchor under all intra (AI) configuration.

Keywords: Versatile video coding · convolutional neural network · compression artifact removal · cyclic DWT · in-loop filter · subband adaptive · transformer.

1 Introduction

In VVC, there are four types of in-loop filters [3, 10]: Deblocking filter (DBF) and sample adaptive offset (SAO), adaptive loop filter (ALF) and Luma mapping with chroma scaling (LMCS). The operation order of these filters is LMCS \rightarrow DBF \rightarrow SAO \rightarrow ALF. LMCS [19], also known as in-loop reshaping, is a technology designed for high dynamic range (HDR) and standard dynamic range (SDR) videos. LMCS consists primarily of two parts: (1) Luminance in-loop mapping based on adaptive piecewise linear models (LM); (2) Chroma scaling based on

This work was supported by the National Natural Science Foundation of China (No. 62111540272).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15326, pp. 202–217, 2025.
https://doi.org/10.1007/978-3-031-78395-1_14

luminance (CS). The key aspect of LMCS lies in determining luminance mapping functions and chroma scaling factors directly relevant to the characteristics of the video source. LM operates at the pixel level, aiming to enhance video encoding efficiency by fully utilizing luma value range and photoelectric conversion property. Conversely, CS operates at the chroma block level, intending to compensate for the influence of luma signal mapping on chroma signals. The primary purpose of DBF [22] is to eliminate block artifacts caused by transformation, quantization, and motion compensation in inter-frame prediction. DBF adaptively decides whether to apply filtering at different block boundaries, the filtering strength, and the maximum filtering length. That is, DBF applies strong filtering to the discontinuous boundaries in smooth regions, weak filtering to the areas with rich textures, or even no filtering at all. The DBF processing begins with horizontal filtering of vertical boundaries across the entire frame, followed by vertical filtering of horizontal boundaries. SAO [6, 7, 16] aims to address the ringing artifacts caused by significant loss of high-frequency information during the quantization process. SAO suppresses ringing artifacts in the pixel domain by classifying reconstructed values, providing negative compensation for multiple peaks and positive compensation for valleys. Therefore, the key aspect of SAO lies in the classification of reconstructed pixels. Similar to the high efficiency video coding (HEVC), SAO in VVC includes two compensation modes of edge offset (EO) and band offset (BO), both applied at the CTB level. ALF [26] technology includes luma ALF, chroma ALF, and cross-component ALF based on Wiener filtering principle. By establishing Wiener-Hopf equations using the original image information and reconstructed image information, a series of filter coefficients with minimum mean square error are solved. ALF aims to reduce the decoding errors effectively, thus improving PSNR values.

In recent years, deep learning technology has made significant strides in the field of image and video processing. Thanks to the outstanding learning capabilities of neural networks, these technologies are now capable of efficiently handling complex image and video data, demonstrating outstanding performance across numerous tasks. The Joint Video Exploration Team (JVET) has not only continued to optimize existing coding tools within the traditional hybrid coding framework during the development of the H.266/VVC standard, but has also extended its focus on the recent and highly anticipated neural network technologies. JVET aims to leverage the sophistication of neural networks for further enhancing the compression efficiency and visual quality in video coding. Li *et al.* [14] proposed a convolutional neural network (CNN) filter to replace existing filters in VTM. This method utilized the reconstructed information, including partition and prediction information, as auxiliary inputs, and trained on augmented training data with the combined MAD and MSE losses. Extensive experiments confirmed the effectiveness of the CNN filter in removing compression artifacts. Nasiri *et al.* [21] introduced a CNN-based video quality assessment method for image processing in VVC. This method was applied to intraframe and interframe coding, which leverages the prediction information to further enhance performance. Bordes *et al.* [5] improved the performance of SAO using neural networks while

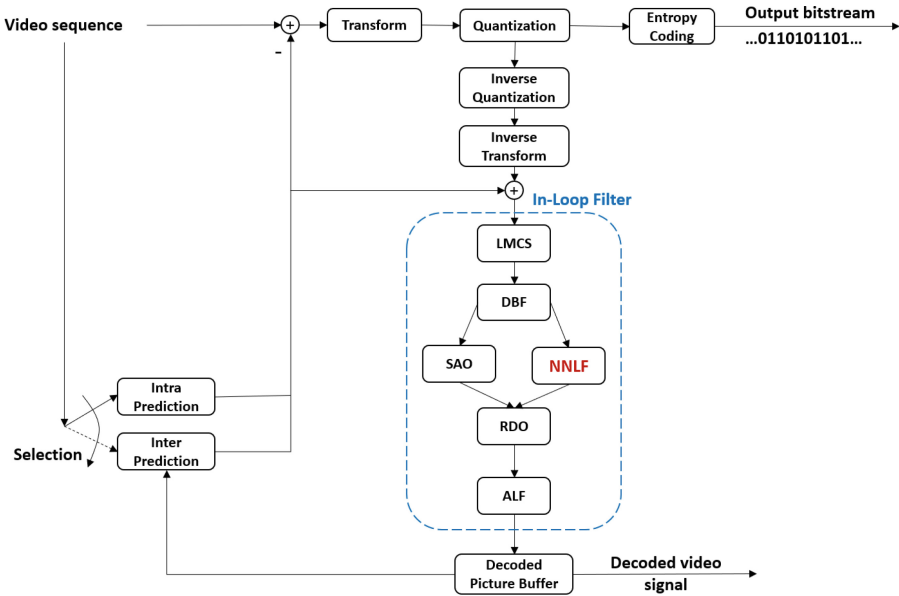


Fig. 1. Whole framework of the VVC encoder with the NN-based in-loop filter (NNLF). LMCS: Luma mapping with chroma scaling. DBF: Deblocking filter. SAO: Sample adaptive offset. RDO: Rate distortion optimization. ALF: Adaptive loop filter. The proposed DWT-SALF is inserted into NNLF of the VVC encoder.

maintaining its fundamental principles, replacing the original reconstructed pixel classification method. In JVET-T0069 [23], a neural network-based model was proposed for subjective optimization rather than objective metric-based optimization, which was operated between DBF and SAO. In the filtering scheme proposed in JVET-AA0088[27], a single-model neural network-based loop filter (NNLF) was proposed, handling luma and chroma information simultaneously. This NNLF took the reconstructed pixels before DBF as inputs, and the output was weighted fused with the SAO filter output as input to the ALF filter. Furthermore, in JVET-AD0380[2], a high-performance operation point (HOP) in-loop filter was proposed as a unified neural network-based loop filter in JVET.

Discrete wavelet transform (DWT) and convolutional neural network (CNN) are two widely used technologies in the fields of image processing and computer vision. DWT is an effective signal processing tool that can decompose images or videos into sub-bands of different scales such as low-frequency and high-frequency sub-bands. These sub-bands contain different frequency information and details of the image or video. CNN has powerful feature extraction and learning capabilities, and is especially good at processing image and video-related tasks. Introducing DWT into CNN can capture the local and global features of images and videos while retaining the multi-scale information of images and videos. Yao *et al.* [29] proposed a new wavelet visual transformer (Wave-ViT),

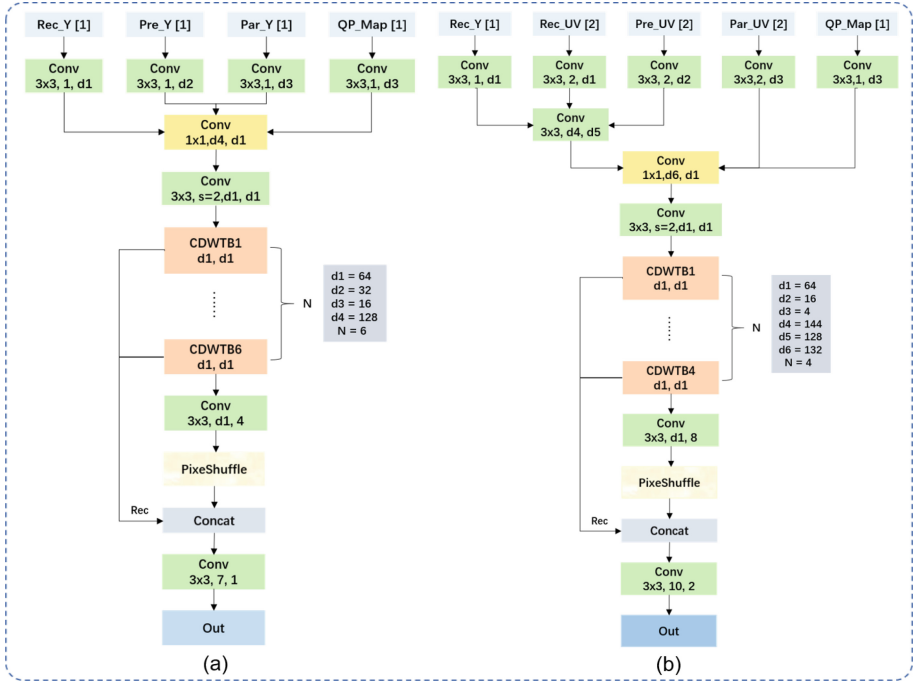


Fig. 2. Network architecture of the proposed DWT-SALF. (a) Luma model. (b) Chroma model. In each block, [Conv, s , $n \times n$, d_{in} , d_{out}] represents convolution, stride (default is 1), convolution kernel size, number of input channels, and number of output channels, respectively. The chroma model takes the luma reconstructed frame to guide and supplement chroma channels (UV). Rec stands for the reconstructed frame, Pre stands for the predicted frame, Par stands for the partition map, and QP_Map stands for the QP map. [1] indicates a single channel, while [2] indicates two channels.

which unified reversible downsampling with wavelet transform and self-attention learning. This solution implements self-attention learning for lossless downsampling of keys/values, helping to pursue a better trade-off between efficiency and accuracy. This avoids information loss caused by downsampling, especially high-frequency components in features such as textures and details. Liu *et al.* [17] proposed a multi-level wavelet CNN (MWCNN) model to expand the receptive field for better trade-off between performance and efficiency. MWCNN is based on U-Net architecture and consists of shrinking subnets and expanding subnets. In the shrinking subnet, DWT is utilized to replace each pooling operation. Since DWT is reversible, it is guaranteed that all information can be preserved by this downsampling scheme. In addition, DWT can capture the frequency and location information of feature maps, which may help preserve detailed textures. In the extended subnetwork, the inverse discrete wavelet transform (IDWT) is used to upsample the low-resolution feature map to the high-resolution feature map. Krishnaraj *et al.* [11] proposed a model that combined DWT and CNN

for real-time image compression in IoUT environment. DWT is responsible for decomposing images into multi-scale sub-bands and capturing local and global features, while CNN learns these features and generates compressed representations to achieve efficient data compression and real-time performance. Qin *et al.* [24] introduced DWT into the attention mechanism. They viewed channel attention from the frequency domain perspective to make up for the shortcomings of insufficient feature information in existing channel attention. They extended global average pooling (GAP) to a more general representation form, namely the 2-dimensional discrete cosine transform (DCT), by introducing more frequencies to make full use of information. VVC is a block-based video coding framework, and inevitably causes compression artifacts such as block artifacts, ringing artifacts, and color distortion, thereby reducing the visual quality and viewing experience. Moreover, quantization techniques are commonly employed during encoding to reduce the amount of video data, which often results in the loss of high-frequency component, which is closely related to edges and textures of an image. The loss of them causes the decoded image to appear blurry and exhibit noticeable block effects.

In this paper, we propose a subband adaptive NNLF in VVC using cyclic DWT, named DWT-SALF. DWT can decompose images or videos into sub-bands of different scales such as low-frequency and high-frequency subbands. Since these sub-bands contain different frequency information in images and videos, the introduction of DWT into CNN can perform subband adaptive processing and learning from images and videos, thereby enhancing the network's ability of capturing features. DWT-SALF combines DWT with CNN to restore high-frequency component and enhance the quality of decoded images. DWT-SALF consists of 6 cyclic DWT blocks (CDWTBs) in luma and 4 CDWTBs in chroma designed to extract important feature information from input images. Each CDWTB includes two residual blocks, a channel attention mechanism, and a cyclic DWT enhancement block (CDWTEB). CDWTEB utilizes DWT to decompose the input feature map into high-frequency subbands and low-frequency subbands. Li *et al.* [12] explored the impact of CNN and transformer on performance from the frequency perspective, finding that transformer excels in capturing low-frequency information but are ineffective in capturing high-frequency features. Therefore, we employ transformer to process low-frequency subband while using CNN to process high-frequency subbands. To increase the network depth and receptive field without adding parameters, CDWTEB is cyclically used within CDWTB, while both the high-frequency and low-frequency branches in CDWTEB are also cyclically utilized twice. We have integrated DWT-SALF into VTM-11.0_NNVC-2.0, which achieves average gains of $\{-8.10\%(Y), -21.19\%(U), -22.28\%(V)\}$ over the VTM-11.0_NNVC-2.0 anchor under AI configuration. Fig. 1 illustrates the whole framework of the VVC encoder with the NN-based in-loop filter (NNLF). The proposed DWT-SALF is inserted into NNLF of the VVC encoder.

Compared with existing methods, main contributions of this paper are summarized as follows:

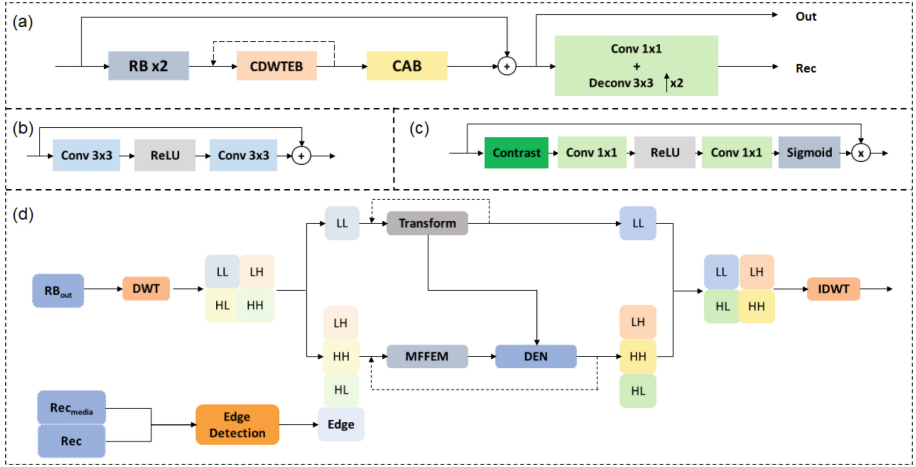


Fig. 3. Network architecture of the cyclic DWT block (CDWTB) in DWT-SALF. (a) CDWTB. (b) Residual block (RB). (c) Channel attention block (CAB) [15]. (d) Cyclic DWT enhancement block (CDWTEB). RB_{out} represents the output of the residual block (RB), Rec_{media} represents the reconstruction output of the previous CDWTB, and Rec represents the input reconstructed frame.

- We propose subband adaptive neural network based in-loop filter for VVC using cyclic DWT, named DWT-SALF. Since DWT decomposes an image into high-frequency subbands and low-frequency subbands, DWT-SALF takes advantages of subband adaptive learning to enhance the quality of decoded frames in VVC.
- DWT-SALF combines CNN and transformer in a DWT-based framework to take each own advantage. Compared to CNN, transformer is effective in capturing low-frequency features but has limited ability of learning high-frequency ones. Thus, we utilize transformer to handle the low-frequency subband and CNN to treat the high-frequency subbands. Moreover, the high-frequency subbands are further enhanced with the guidance of the processed low-frequency subband.
- DWT-SALF adopts a cyclic DWT block (CDWTB) as the basic block to increase the network depth and receptive field without increasing parameters. In CDWTB, CDWTEB and both high-frequency and low-frequency branches inside CDWTEB are cyclically used twice to enhance the features.
- DWT-SALF employs a latent edge map as auxiliary information to enrich features for the high-frequency subbands. The latent edge map is obtained by applying simple edge detection to the input reconstructed frame and the reconstructed output of the previous CDWTB.

2 Proposed Method

As illustrated in Fig. 2, the proposed DWT-SALF consists of luma and chroma models. Both the luma and chroma models have the network architecture of head, backbone, and reconstruction. The head in the luma model includes four inputs: the reconstructed frame (Red_Y [1]), predicted frame (Pre_Y [1]), partition map (Par_Y [1]), and QP map (QP_Map [1]). Each input undergoes feature extraction with a 3×3 convolutional layer, followed by feature fusion and dimensionality reduction using a 1×1 convolutional layer. Finally, the resolution is reduced by a 3×3 convolutional layer with a stride of 2 to reduce computational complexity. The head of the chroma model includes five inputs: the luma reconstructed frame (Rec_Y [1]), chroma reconstructed frame (Pre_UV [2]), chroma predicted frame (Pre_UV [2]), chroma partition map (Par_UV [2]), and QP map (QP_map [1]). Since the luma reconstructed frame contains rich information compared to the chroma reconstructed frame, we use the luma reconstructed frame to guide and supplement chroma channels (UV). In DWT-SALF, the backbones of both luma and chroma models use cyclic DWT block (CDWTB) as the basic block. The luma model contains 6 CDWTBs, while the chroma model contains 4 CDWTBs. Since the chroma channels are simpler than the luma channel, they have the small number of CDWTBs.

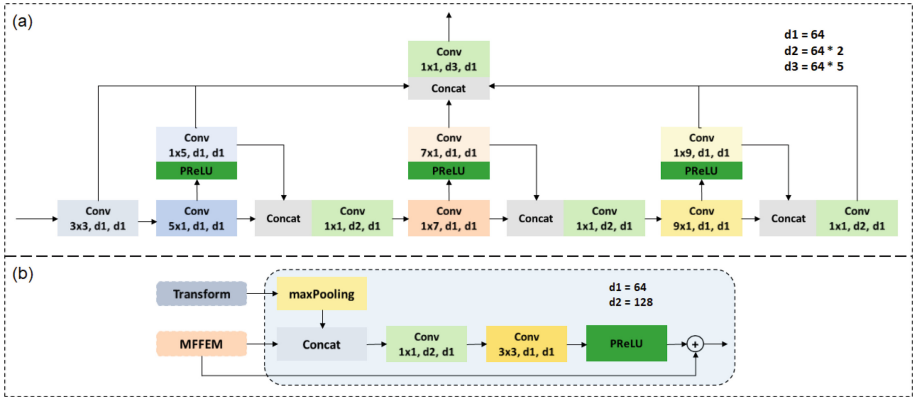


Fig. 4. Network architectures of two core modules in DWT-SALF. (a) Multi-scale feature fusion enhancement module (MFFEM). (b) Detail enhancement module (DEM).

2.1 Cyclic DWT Block

Fig. 3(a) illustrates the specific process of the cyclic DWT block (CDWTB). CDWTB consists of two residual blocks (RB), a cyclic DWT enhancement block (CDWTEB), and a channel attention block (CAB). The input of the i -th CDWTB is derived from the $(i-1)$ -th CDWTB. The input feature maps

undergo feature extraction through two RBs, followed by CDWTEB to enhance high-frequency and low-frequency features separately. CDWTEB is cyclically used twice, where the output of the first cycle serves as the input to the second cycle. This dual-cycle utilization of CDWTEB deepens the network and enlarges the receptive field without increasing parameters. Subsequently, CAB is utilized to enhance model performance at the channel level, thereby improving overall model performance. To reduce the training complexity and mitigate gradient vanishing, CDWTB utilizes residual connections. CDWTB has two outputs: one is the output of the residual connection, and the other is the reconstructed frame generated by applying 1x1 convolution and transposed convolution to the result of the residual connection.

Table 1. BD-rate of the first-stage model for DWT-SALF over VTM-11.0_NNVC-2.0 in AI configuration. DWT-SALF is embedded into VTM-11.0_NNVC-2.0 where the QP distance is set to 5. For training set, the input QP is {22, 27, 32, 37, 42}, while its corresponding label QP is {17, 22, 27, 32, 37}.

Class	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.09%	-21.36%	-23.63%
Class A2	-7.25%	-21.39%	-21.73%
Class B	-6.94%	-20.22%	-20.29%
Class C	-6.87%	-15.23%	-18.17%
Class E	-9.61%	-25.06%	-23.76%
Overall	-7.45%	-20.30%	-21.19%
Class D	-6.56%	-15.69%	-17.45%

2.2 Cyclic DWT Enhancement Block

The network structure of the cyclic DWT enhancement block (CDWTEB) is shown in Fig. 4(d). First, the RB output is decomposed into high-frequency subbands and low-frequency subbands through DWT. Li *et al.*[12] found that transformers excel in capturing low-frequency information but perform less effectively in capturing high-frequency information. The unique feature of transformer is a global attention mechanism, which can comprehensively consider the correlation between any pixels in the image and takes advantage of understanding and inferring the connections between distant elements in an image. In contrast, CNN has a fixed receptive field, while transformer can more flexibly adapt to various complex image scenes and adjust the focus of attention according to the content. Considering the advantages of transformer, we use a transformer model proposed by Zamir *et al.*[30] to process and enhance features in low-frequency subband. However, due to the poor ability of transformer in capturing high-frequency information, we use CNN to process high-frequency subbands

and propose a multi-scale feature fusion enhancement module (MFFEM). To supplement insufficient high-frequency features, we use Sobel edge extraction to extract edge information from the reconstruction output of the previous CDWTB $\text{Rec}_{\text{media}}$ and the input reconstructed frame Rec . The latent edge map is put into MFFEM together with the high-frequency subbands to obtain the enhanced high-frequency information. Since the low-frequency information contains relatively rich information, we further utilize the enhanced low-frequency information to enhance the high-frequency information. The enhanced low-frequency information and high-frequency information are put into the detail enhancement module (DEM) to further enhance the high-frequency feature. To increase the network depth and fully extract features without increasing the number of parameters, the high-frequency branch and the low-frequency branch are recycled twice in CDWTEB. Finally, the enhanced high-frequency feature and low-frequency feature are fused through IDWT.

Table 2. BD-rate of the second-stage model for DWT-SALF over VTM-11.0_NNVC-2.0 in AI configuration. DWT-SALF is embedded into VTM-11.0_NNVC-2.0 where the QP distance is set to 10. For training set, the input QP is {22, 27, 32, 37, 42}, while its corresponding label QP is {17, 22, 27, 32, 37}.

Class	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.27%	-19.76%	-22.19%
Class A2	-7.55%	-21.45%	-22.21%
Class B	-7.25%	-20.51%	-20.14%
Class C	-7.24%	-15.72%	-18.84%
Class E	-10.09%	-24.57%	-23.36%
Overall	-7.78%	-20.15%	-21.08%
Class D	-6.85%	-16.27%	-18.05%

2.3 Multi-scale Feature Fusion Enhancement Module

The network structure of the multi-scale feature fusion enhancement module (MFFEM) is shown in Fig. 4(a). MFFEM significantly expands its receptive field and strengthens feature extraction capability by introducing convolutional kernels of different sizes. In addition to standard 3x3 and 1x1 convolutional kernels, MFFEM also utilizes larger-sized kernels such as 5x5, 7x7, and 9x9 to capture rich feature. Although larger kernels provide deeper levels of details, they inevitably increase computational complexity. To mitigate this complexity, the 5x5, 7x7, and 9x9 convolutions within MFFEM are decomposed [25], thus reducing the computational cost. The strategy of using multi-scale convolutional kernels not only enhances the model’s semantic understanding of input data but also strengthens its ability of capturing multi-scale features, thus significantly improving the overall feature extraction performance of the model.

2.4 Detail Enhancement Module

The network structure of the detail enhancement module (DEM) is illustrated in Fig. 4(b). Low-frequency information contains richer content than high-frequency information. Therefore, we utilize the enhanced low-frequency information to further enhance the high-frequency information. The DEM module first highlights detailed features such as edges and textures in the low-frequency information through max-pooling operations, and then concatenates this with the enhanced high-frequency information along the channel dimension. Subsequently, 1x1 and 3x3 convolutions are applied to the concatenated features to reduce dimensionality and fuse them, thereby obtaining further enhanced high-frequency information. DEM fully leverages the rich content in the low-frequency information, and is combined with the high-frequency information to effectively enhance details and quality of an image.

Table 3. BD-rate of the third-stage model for DWT-SALF over VTM-11.0_NNVC-2.0 in AI configuration. DWT-SALF is embedded into VTM-11.0_NNVC-2.0 where the QP distance is set to 15. For training set, the input QP is {22, 27, 32, 37, 42}, while its corresponding label QP is {17, 22, 27, 32, 37}.

Class	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.61%	-21.64%	-23.76%
Class A2	-7.81%	-22.88%	-23.68%
Class B	-7.58%	-21.30%	-21.31%
Class C	-7.53%	-15.84%	-19.51%
Class E	-10.53%	-25.98%	-24.71%
Overall	-8.10%	-21.19%	-22.28%
Class D	-7.11%	-16.52%	-18.68%

3 Experimental Results

For the experiments, we trained two models of luma and chroma as shown in Fig. 2. We embedded the trained models into VTM-11.0_NNVC-2.0 and performed evaluation under the Common Test Conditions (CTC) in JVET [18]. We compared the results with VTM-11.0_NNVC-2.0 anchor. The specific configurations and results are given below.

3.1 Experimental Setting

We implemented DWT-SALF on the PyTorch 1.9.0 framework and trained it on a PC equipped with an NVIDIA GeForce GTX 4090 GPU. We adopted the progressive learning based on QP distance proposed by Zhang *et al.* [32] to train

Table 4. BD-rate of DWT-SALF over VTM-11.0_NNVC-2.0 anchor in RA configuration. The trained models are embedded into VTM-11.0_NNVC-2.0 and the QP distance is set to 10. The input QP of the training set is {22, 27, 32, 37, 42}, while its corresponding label QP is {12, 17, 22, 27, 32}.

Class	Y-PSNR	U-PSNR	V-PSNR
Class A1	-9.22%	-19.37%	-20.49%
Class A2	-10.20%	-22.23%	-23.86%
Class B	-8.87%	-24.15%	-20.85%
Class C	-9.07%	-16.60%	-17.10%
Overall	-9.26%	-20.80%	-20.38%
Class D	-10.60%	-16.88%	-16.44%

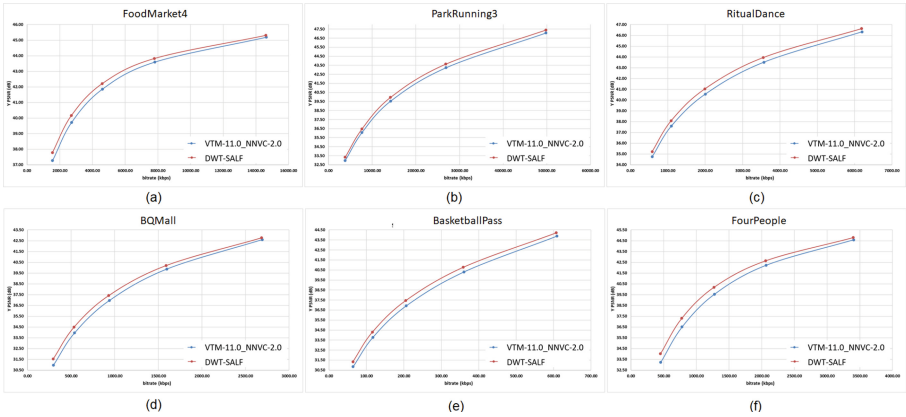


Fig. 5. RD curves by VTM-11.0_NNVC-2.0 and DWT-SALF under the AI configuration. x -axis: Bitrate (kbps). y -axis: Y PSNR (dB). (a) *FoodMarket4* in Class A1. (b) *ParkRunning3* in Class A2. (c) *RitualDance* in Class B. (d) *BQMall* in Class C. (e) *BasketballPass* in Class D. (f) *FourPeople* in Class E.

DWT-SALF. The QP distance-based training strategy consists of three stages with 80 epochs in the first stage, 60 epochs in the second stage, and 60 epochs in the third stage. In each stage of training, the final 5 epochs use L2 Loss, while the rest use L1 Loss. We set the batch size to 32, initialized the learning rate to $1e-4$, gradually decreasing it to $1e-5$. We utilized the Adam optimizer for model optimization. We used the DIV2K[1] and BVI_DVC[20] datasets to train DWT-SALF with total 1000 images (800 from DIV2K and 200 from BVI_DVC). After compression by VTM-11.0_NNVC-2.0, all images were cropped into blocks of size 144×144 (luma) and 72×72 (chroma). To augment the dataset, we applied random vertical and horizontal flipping operations to the training data. Finally, we embedded the trained models into VTM-11.0_NNVC-2.0 and tested it using

video sequences specified in CTC [18], and evaluated the performance in terms of BD-rate [4].

3.2 Visual Comparison

We embedded the trained model into VTM-11.0_NNVC-2.0 and conducted tests under CTC [18]. We tested the performance of DWT-SALF at five QP {22, 27, 32, 37, 42} in both AI and RA configurations. The results by the first stage model to the third stage model under AI configuration are provided in Tables 1, 2, and 3 in AI configuration, while the results under RA configuration are provided in Table 4. In the RA configuration, considering the training time, the QP distance is set to 10 and DWT-SALF is trained in one stage. The rate distortion (RD) curves of some classes in AI configuration are shown in Fig. 5. Fig. 6 provides visual comparison to demonstrate the effectiveness of DWT-SALF. In the figure, we provide the uncompressed frame, the compressed frame by the VTM-11.0_NNVC-2.0 anchor, and the compressed frame by DWT-SALF. It is obvious that DWT-SALF successfully reconstructs the textures and details in the frames and outperforms the VTM-11.0_NNVC-2.0 anchor in visual quality.

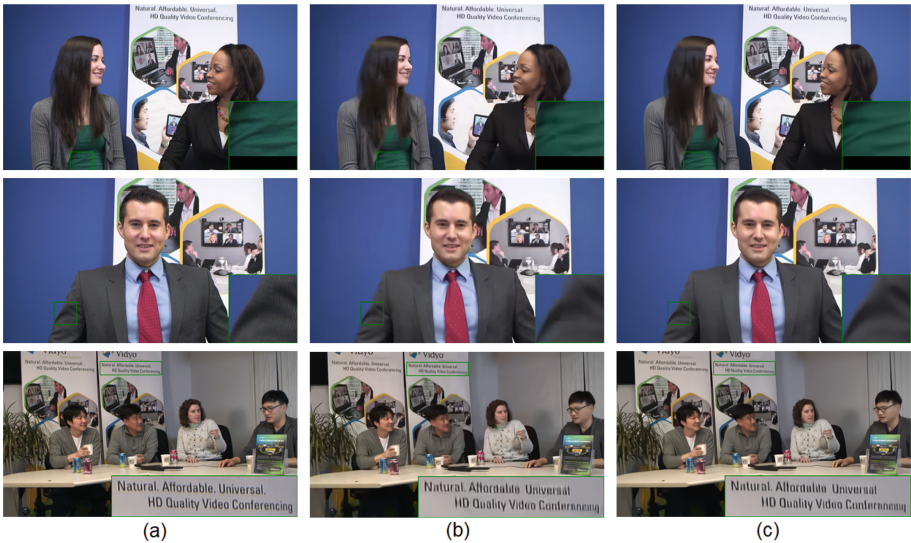


Fig. 6. Visual comparison between the VTM-11.0_NNVC-2.0 anchor and DWT-SALF. (a) Uncompressed frames. (b) VTM-11.0_NNVC-2.0 anchor. (c) DWT-SALF. Top to bottom: *KristenAndSara*, *Johnny*, and *FourPeople* in Class E. We obtain the results at QP 42 under the AI configuration.

3.3 Ablation Study

In the ablation study, we explored the impact of DWT on the performance. In the above-mentioned experiments, we used DWT to decompose the frame into low-frequency and high-frequency information, which were then processed separately using transformer and CNN. In this ablation experiment, we replaced DWT with convolutional layers of stride 2 and increased the number of channels by four times. Subsequently, we proportionally split the feature maps and fed them into transformer and MFFEM for processing. Finally, the processed feature map was concatenated along the channel dimension, and then PixelShuffle was used instead of IDWT to restore the channel number and resolution. The results of the ablation experiment are shown in Table 5. Compared to Table 3 that used DWT to decompose features into low-frequency and high-frequency components, the results without DWT show a decrease of 0.04%, 0.21%, 0.36% in BD-rate for {Y, U, V} channels, which demonstrates the effectiveness of DWT for feature decomposition and processing with transformer and CNN.

Table 5. Ablation study on DWT. After training, the third-stage model was embedded into VTM-11.0_NNVC-2.0 and the test results were obtained in AI configuration with a QP distance set to 15.

Class	Y-PSNR	U-PSNR	V-PSNR
Class A1	-7.63%	-21.27%	-22.78%
Class A2	-7.76%	-22.49%	-23.68%
Class B	-7.49%	-21.21%	-20.96%
Class C	-7.44%	-16.17%	-19.10%
Class E	-10.55%	-25.24%	-24.64%
Overall	-8.06%	-20.98%	-21.92%
Class D	-6.99%	-16.10%	-18.24%

Table 6. Comparison of BD-rate and complexity among DWT-SALF and other JVET contributions over VTM-11.0_NNVC-2.0 anchor in AI configuration.

Method	Y-PSNR	U-PSNR	V-PSNR	Parameters	KMAC/pixel
Z0091[28]	-6.50%	-14.89%	-15.98%	1.9M(Luma+Chroma)	485K(Luma+Chroma)
AA0111[13]	-7.26%	-20.14%	-20.56%	1.56M(Luma) 1.56M(Chroma)	682K(Luma) 682K(Chroma)
AB0090[31]	-7.08%	-12.46%	-12.75%	0.78M(Luma+chroma)	200K(Luma+chroma)
AC0118[33]	-7.49%	-20.60%	-21.18	1.56M(Luma) 1.56M(Chroma)	682K(Luma) 682K(Chroma)
AE0191[8]	-7.78%	-18.81%	-19.98%	1.45M(Luma+chroma)	477K(Luma+chroma)
AF0041[9]	-7.91%	-18.69%	-20.23%	1.45M(Luma+chroma)	477K(Luma+chroma)
DWT-SALF	-8.10%	-21.19%	-22.28%	2.69M(Luma) 1.84M(Chroma)	620K(Luma) 441K(Chroma)

3.4 Comparison With Other In-Loop Filters

We have selected a number of proposals from the latest JVET meetings on NN-based in-loop filters (NNLFs) and compared them with DWT-SALF under AI configuration. These proposals are JVET-Z0091 [28], JVET-AA0111 [13], JVET-AB0090 [31], JVET-AC0118 [33], JVET-AE0191 [8], and JVET-AF0041 [9]. Table 6 shows comparison of BD-rate and complexity among DWT-SALF and other JVET contributions over VTM-11.0_NNVC-2.0 anchor in AI configuration. Although DWT-SALF does not perform the best in terms of parameter amount and computational complexity (KMAC/pixel), DWT-SALF achieves the highest gain across the Y, U, and V channels with moderate complexity.

4 Conclusions

In this paper, we have proposed a subband adaptive NNLF for VVC using cyclic DWT, named DWT-SALF. We have used DWT to decompose the image into high-frequency subbands and low-frequency subbands. Compared to CNNs, transformers are more effective in capturing low-frequency information but have limited ability of learning high-frequency representation. Thus, we have utilized transformer to handle the low-frequency branch and CNN to treat the high-frequency branch, i.e. subband adaptive learning. We have further enhanced the high-frequency feature using the processed low-frequency feature. Moreover, we have employed a dual-cycle strategy inside both CDWTB and CDWTEB to increase the network depth and receptive field without increasing parameters. Experimental results demonstrate that DWT-SALF achieves average BD-rate gains of $\{-8.10\%$ (Y), -21.19% (U), -22.28% (V) $\}$ over the VTM-11.0_NNVC-2.0 anchor in AI configuration and outperforms state-of-the-art methods in terms of visual quality and quantitative measurements.

Our future work includes optimizing the network architecture to reduce parameters and complexity while maintaining the performance.

References


1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proc. IEEE CVPR Workshops. pp. 126–135 (2017)
2. Alshina, E., Galpin, F.: Bog report on nn-filter design unification. JVET-AD0380 (April 2023)
3. Amruthavalli, P.L., Nalluri, P.: A review on in-loop filters for hevc and vvc video coding standards. In: Proceedings of the International Conference on Advanced Computing and Communication Systems. vol. 1, pp. 997–1001 (2022)
4. Bjøntegaard, G.: Calculation of average psnr differences between rd-curves. ITU SG16 Doc. VCEG-M33 (2001)
5. Bordes, P., Galpin, F., Dumas, T., Nikitin, P.: Revisiting the sample adaptive offset post-filter of vvc with neural-networks. In: Proc. PCS. pp. 1–5 (2021)
6. Fu, C.M., Alshina, E., Alshin, A., Huang, Y.W., Chen, C.Y., Tsai, C.Y., Hsu, C.W., Lei, S.M., Park, J.H., Han, W.J.: Sample adaptive offset in the hevc standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1755–1764 (2012)

7. Fu, C.M., Chen, C.Y., Huang, Y.W., Lei, S.: Sample adaptive offset for hevc. In: Proc. IEEE MMSP. pp. 1–5 (2011)
8. Galpin, F., Eadie, S., Li, Y., Wang, L., Xie, Z., Rusanovskyy, D., Li, Y., Chang, R., Li, J., Alshina, E.: Ahg11 - ee1-0 high operation point model. JVET-AE0191 (July 2023)
9. Galpin, F., Rusanovskyy, D., Li, Y., Wang, L., Xie, Z., Li, Y., Chang, R., Li, J., Alshina, E.: Ahg11: Hop full results. JVET-AF0041 (October 2023)
10. Karczewicz, M., Hu, N., Taquet, J., Chen, C.Y., Misra, K., Andersson, K., Yin, P., Lu, T., François, E., Chen, J.: Vvc in-loop filters. IEEE Trans. Circuits Syst. Video Technol. **31**(10), 3907–3925 (2021)
11. Krishnaraj, N., Elhoseny, M., Thenmozhi, M., Selim, M.M., Shankar, K.: Deep learning model for real-time image compression in internet of underwater things (iout). J. Real-Time Image Proc. **17**(6), 2097–2111 (2020)
12. Li, A., Zhang, L., Liu, Y., Zhu, C.: Feature modulation transformer: Cross-refinement of global representation via high-frequency prior for image super-resolution. In: Proc. IEEE ICCV. pp. 12514–12524 (2023)
13. Li, Y., Zhang, K., Li, J., Zhang, L., Wang, H., Galpin, F.: Ee1-1.6: Deep in-loop filter with fixed point implementation. JVET-AA0111 (July 2022)
14. Li, Y., Zhang, L., Zhang, K.: Convolutional neural network based in-loop filter for vvc intra coding. In: Proc. IEEE ICIP. pp. 2104–2108 (2021)
15. Li, Z., Liu, Y., Chen, X., Cai, H., Gu, J., Qiao, Y., Dong, C.: Blueprint separable residual network for efficient image super-resolution. In: Proc. IEEE CVPR. pp. 833–843 (2022)
16. Lim, W.Q., Schwarz, H., Marpe, D., Wiegand, T.: Post sample adaptive offset for video coding. In: Proc. PCS. pp. 1–5 (2019)
17. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-cnn for image restoration. In: Proc. IEEE CVPR Workshops. pp. 773–782 (2018)
18. Liu, S., Segall, A., Alshina, E., Liao, R.L.: Common test conditions and evaluation procedures for neural network-based video coding technology. JVET-W2016 (August 2021)
19. Lu, T., Pu, F., Yin, P., McCarthy, S., Husak, W., Chen, T., Francois, E., Chevance, C., Hiron, F., Chen, J., et al.: Luma mapping with chroma scaling in versatile video coding. In: Proc. DCC. pp. 193–202 (2020)
20. Ma, D., Zhang, F., Bull, D.R.: Bvi-dvc: A training database for deep video compression. IEEE Trans. Multimedia **24**, 3847–3858 (2021)
21. Nasiri, F., Hamidouche, W., Morin, L., Dhollande, N., Cocherel, G.: Model selection cnn-based vvc quality enhancement. In: Proc. PCS. pp. 1–5 (2021)
22. Norkin, A., Bjontegaard, G., Fuldseth, A., Narroschke, M., Ikeda, M., Andersson, K., Zhou, M., Van der Auwera, G.: Hevc deblocking filter. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1746–1754 (2012)
23. Ouyang, T., Liu, F., Zhu, H., Chen, Z., Xu, X., Liu, S.: Ahg11: Ssim based cnn model for in-loop filtering. JVET-T0069 (October 2020)
24. Qin, Z., Zhang, P., Wu, F., Li, X.: Fcanet: Frequency channel attention networks. In: Proc. IEEE ICCV. pp. 783–792 (2021)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proc. IEEE CVPR. pp. 2818–2826 (2016)
26. Tsai, C.Y., Chen, C.Y., Yamakage, T., Chong, I.S., Huang, Y.W., Fu, C.M., Itoh, T., Watanabe, T., Chujoh, T., Karczewicz, M., et al.: Adaptive loop filtering for video coding. IEEE Journal of Selected Topics in Signal Processing **7**(6), 934–945 (2013)

27. Wang, L., Lin, S., Xu, X., Liu, S., Galpin, F.: Ee1-1.5: Neural network based in-loop filter with a single model. JVET-AA0088 (July 2022)
28. Wang, L., Xu, X., Liu, S., Galpin, F.: Ee1-1.2: neural network based in-loop filter with a single model. JVET-Z0091 (April 2022)
29. Yao, T., Pan, Y., Li, Y., Ngo, C.W., Mei, T.: Wave-vit: Unifying wavelet and transformers for visual representation learning. In: Proc. ECCV. pp. 328–345 (2022)
30. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proc. IEEE CVPR. pp. 5728–5739 (2022)
31. Zhang, H., Jung, C., Liu, Y., Li, M.: Ee1-1.3 related: Lightweight and efficient cnn in-loop filter. JVET-AB0090 (October 2022)
32. Zhang, H., Jung, C., Liu, Y., Li, M.: Lightweight cnn-based in-loop filter for vvc intra coding. In: Proc. IEEE ICIP. pp. 1635–1639 (2023)
33. Zhou, C., Lv, Z., Zhang, J.: Ee1-1.8: Qp-based loss function design for nn-based in-loop filter. JVET-AC0118 (January 2023)



Optimal Time Sampling in Physics-Informed Neural Networks

Gabriel Turinici^(✉) 

CEREMADE, Université Paris Dauphine - PSL, Place du Marechal de Lattre de
Tassigny, Paris 75016, France
Gabriel.Turinici@dauphine.fr
<https://turinici.com>

Abstract. Physics-informed neural networks (PINN) is an extremely powerful paradigm used to solve equations encountered in scientific computing applications. An important part of the procedure is the minimization of the equation residual which includes, when the equation is time-dependent, a time sampling. It has been argued in the literature that the sampling need not be uniform but should overweight initial time instants, but no rigorous explanation was provided for this choice. In the present work we take some prototypical examples and, under standard hypotheses concerning neural network convergence, we show that the optimal time sampling follows a (truncated) exponential distribution. In particular we explain when it is best to use uniform time sampling and when one should not. The findings are illustrated with numerical examples on a linear equation, Burgers' equation and the Lorenz system.

1 Introduction and literature review

Following their recent introduction in [11], physics-informed neural networks became a powerful tool invoked in scientific computing to numerically solve ordinary (ODE) or partial (PDE) differential equations in physics [9] including high dimensional (e.g. Schrodinger) equations [6], finance [2, 13], control problems [7], data assimilation and so on. As such it became an important framework that leverages the power of neural networks (NN). Even if successful applications are reported for many situations encountered in numerical simulations, however the workings of PINNs are not yet fully optimized and research efforts are nowadays targeted towards improving the output quality or training process, cf. [17] and related works.

We will focus here on time-depending equations that can be formalized as solving

$$\partial_t u(t, x) = \mathcal{F}(u), \tag{1}$$

$$u(0, x) = u_0, \quad \forall x \in \Omega \tag{2}$$

$$u(t, x) = u_b(t, x), \quad \forall x \in \partial\Omega, \forall t \in [0, T], \tag{3}$$

where \mathcal{F} is an evolution operator (see below for examples), u is the unknown, u_0 the initial condition, Ω the spatial domain and $u_b(t, x)$ the conditions at the boundary $\partial\Omega$ of Ω ; this can be supplemented with additional measured quantities or physical information in data assimilation settings (not used here). We will not investigate in this work the difference between weak and strong formulations so we suppose (1) is true pointwise with classical time and partial derivatives (classical solutions). Note that when Ω is a discrete set (1) becomes a ordinary differential equation (ODE) and in this case ∂_t is to be replaced by d/dt .

In its simplest form, the PINN approach constructs a neural network \mathcal{U} indexed by parameters θ mapping the input $(t, x) \in [0, T] \times \Omega$ to $\mathcal{U}_\theta(t, x) \in \mathbb{R}$ that will stand for the (unknown) solution $u(t, x)$ (see figure 1 for an illustration). To ensure that \mathcal{U}_θ is close to u the following functional is minimized with respect to θ by usual means of deterministic or stochastic optimization as is classically done for NNs :

$$L(\theta) := \int_0^T \int_{\Omega} E_\theta(t, x)^2 dx dt + \quad (4)$$

$$c_{ic} \int_{\Omega} (\mathcal{U}_\theta(t, x) - u_0(x))^2 dx + c_{bc} \int_0^T \int_{\partial\Omega} (\mathcal{U}_\theta(t, x) - u_{bc}(t, x))^2 dx dt. \quad (5)$$

Here c_{ic} , c_{bc} are some positive coefficients and $E_\theta(t, x)$ is the error term :

$$E_\theta(t, x) = \partial_t \mathcal{U}_\theta(t, x) - \mathcal{F}(\mathcal{U}_\theta)(t, x). \quad (6)$$

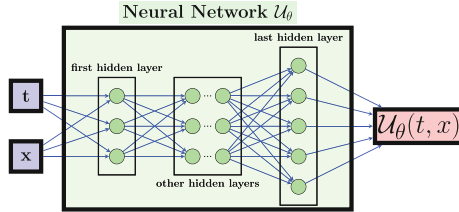


Fig. 1. An illustration of network \mathcal{U}_θ . It takes as input a time t and a space value x and outputs the solution candidate $\mathcal{U}_\theta(t, x)$ for this input couple. The NN is trained so that $\mathcal{U}_\theta(t, x)$ is close to the solution u of (1).

Assuming that the NN is expressive enough i.e., that the true solution u belongs to the set of all possible NN mappings \mathcal{U}_θ then the minimizer of the loss is exactly the solution u . The integral terms of $L(\theta)$ are generally computed through either collocation points or random sampling. The focus of this paper is on the computation of the terms involving the time integral and more precisely we will mostly investigate the term in (4).

The choice of the collocation points has an impact on the efficiency of the PINN result. For instance in [14] the authors argue that adapting the location of

these points by over-weighting areas where E_θ is large will improve the outcome. In another approach, [18] C.L. Wight and J. Zhao propose several adaptive sampling in space and time; among their proposals is the time marching where the time interval $[0, T]$ is divided into segments solved sequentially (their “approach II”); a different approach considers a total span $[0, t]$ that is increased from a small value t to the target value T (“approach I”). Asking the same question, M. Penwarden and co-workers proposed in a very recent work [9] “a stacked-decomposition method” that combines time marching with a form of Schwartz overlapping time domain decomposition method. Investigation of time sampling also lead S. Wang, S. Sankaran and P. Perdikaris [16] to introduce the causality concept where it is recognized that error made earlier in the time interval will escalate to the final time T ; they propose to over-sample points close to 0 and decrease the sampling weight as time progresses. Our contribution lies very much within this line or thought but here we give rigorous insights into several points :

- what is the best functional form for the decrease in sampling weights from 0 to T
- for what problems is this causal sampling likely to give best results and where is it less critical ? In particular are there any situations where it is optimal to under-weight points near 0 and over-weight points near T ?
- how is this related to the optimization procedure used to minimize the loss functional.

The balance of the paper is as follows : in section 2 we introduce notations and give theoretical insights that we illustrate in section 3 with numerical results; we conclude with remarks in section 4.

2 Theoretical setting and results

To explain the sampling / weighting question, we consider here the simplest possible setting, that of a linear ODE : $\text{cardinality}(\Omega) = 1$, $\mathcal{F}(u) = \lambda u$:

$$u'(t) = \lambda u(t), \quad u(0) = u_0. \quad (7)$$

The network \mathcal{U} parameterized by some θ maps any input $t \in [0, T]$ into the value $\mathcal{U}_\theta(t)$. We want the mapping $t \mapsto \mathcal{U}_\theta(t)$ to represent the solution to (7) and in this case the equation error is

$$f_\theta(t) := \mathcal{U}'_\theta(t) - \lambda \mathcal{U}_\theta(t). \quad (8)$$

To simplify again our setting we will not describe the treatment of the initial condition u_0 and instead assume the network outputs some function that already has $\mathcal{U}_\theta(0) = u_0$. Note that in this simple setting one can easily ensure this equality by just shifting the output

$$\mathcal{U}_\theta(t) \mapsto \mathcal{U}_\theta(t) - \mathcal{U}_\theta(0) + u_0. \quad (9)$$

Similar techniques have been used in the literature, see [16, section E]. Then the PINN method prescribes to minimize, with respect to θ , the following loss function :

$$L_\rho(\theta) := \mathbb{E}_{t \sim \mu} f_\theta(t)^2 = \int_0^T f_\theta^2(t) \rho(t) dt. \quad (10)$$

The loss is the second moment of the equation error f_θ introduced in (8). Here time t follows a probability law μ supported in $[0, T]$ and density $\rho(\cdot)$; we write $\mu(dt) = \rho(t)dt$.

The final goal is to obtain a good approximation of the final solution $u(T)$ which is the unknown and the main goal of the procedure. So the real quantity to be minimized is $|u(T) - \mathcal{U}_\theta(T)|$ but this cannot be done directly because $u(T)$ is not known.

2.1 Model for computational resources

To find the solution one uses (stochastic or deterministic) optimization algorithms, most of them derived from the initial proposal of Robbins and Monro [12]) that was called latter Stochastic Gradient Descent. In turn this was followed by a large set of proposals used nowadays in neural network optimization (Nesterov, momentum, Adam, RMSprop, etc). The deterministic counterpart algorithms (gradient descent, BFGS, L-BFGS and so on) appear on the other hand in standard textbooks [10]. These algorithms find the solution in an iterative manner and convergence is ensured only in the limit of infinite iterations. So we never have the exact solution but some approximation of it. Moreover the computational resources are not infinite either so in practice one is limited by the available resources (in wall clock time or in total operations count or in any other metric). In particular, smaller is the absolute value of $f_\theta(\cdot)$ more computational resources are consumed.

To model this cost we refer to general results on the convergence of optimization algorithms. The convergence of the stochastic and deterministic procedures has been analyzed in detail see [4] for a classic textbook and [5, 8] for recent works or self-contained proofs [1, 15]. It was proved that, in general, the convergence to the exact solution occurs at various speeds including quadratic or exponential convergence. The most often the square of the error is of order $O(\frac{1}{n})$ where n is the number of iterations, proportional to the numerical effort. For convenience we will denote from now on the error by $w(t)$ so finally we have that the square of the error is, say, of order $w(t)^2 \sim O(\frac{1}{n})$. So, if we take as a constraint that the total numerical cost is bounded by some $B \geq 0$, the error $w(t)$ will be associated to a cost of order $1/w(t)^2$ so the optimization algorithm will find some error $w(t)$ that satisfies

$$\int_0^T \frac{1}{w(t)^2} dt \leq B. \quad (11)$$

Of course, the exact functional form of (11) is subject to discussion and e.g., when exponential convergence occurs we will rather have

$$\int_0^T -\ln(w(t)^2)dt \leq B. \tag{12}$$

In fact the arguments below apply to both such formulations and to many other also, so for simplicity we will suppose (11) is true.

To conclude, if computational resources are limited by a total amount B we only have access to errors $w(t)$ that satisfy (11) and not better. The question is how to choose w to minimize the final error between the numerical and the exact solution and how does the time sampling enters into this quest.

Some remarks remains still to be made at this juncture: when minimizing some loss functional under computational constraints (11), it may happen that several values give the same loss level and same computational cost; in this case we cannot be sure which one we will get. So, in a prudent stance, we will suppose from now on that

(H Opt) : The computational procedure results in some error level $w(\cdot)$ that minimizes the loss functional under constraint (11). If several errors give the same cost and loss level we will assume the worse one is actually obtained.

2.2 Step 1: overall optimality

We give here a first result that will be an lower bound on the error $|\mathcal{U}_\theta(T) - u(T)|$.

Proposition 1. *Denote*

$$w(t) := \mathcal{U}'_\theta(t) - \lambda \mathcal{U}_\theta(t), \tag{13}$$

and assume that (11) holds true. Then under hypothesis **(H Opt)** the error $|\mathcal{U}_\theta(T) - u(T)|$ is at least equal to

$$\frac{1}{B^{1/2}} \left(\frac{3(e^{2\lambda T/3} - 1)}{2\lambda} \right)^{3/2}, \tag{14}$$

with equality when $w(t)$ is proportional to $e^{-\lambda(T-t)/3}$.

Proof : We deal here with an optimization problem and need to find the minimum value of the error under resources constraints (11). In general this can be formulated as a Euler-Lagrange constraint optimization problem. But in this particular case it can be settled more directly. Let us first write the definition (13) of w as : $\mathcal{U}'_\theta(t) = \lambda \mathcal{U}_\theta(t) + w(t)$. Then, denoting $\delta u(t) = \mathcal{U}_\theta(t) - u(t)$ we can write $\delta u(t)' = \lambda \delta u(t) + w(t)$, or, by using classical formulas for the solution of this equation :

$$|\mathcal{U}_\theta(T) - u(T)| = |\delta u(T)| = \left| \int_0^T e^{\lambda(T-t)} w(t) dt \right|. \tag{15}$$

Of course, the worse case is realized when $w(s)$ is positive and then we will have $|\mathcal{U}_\theta(T) - u(T)| = \int_0^T e^{\lambda(T-t)} w(t) dt$. Use now the Hölder inequality for the functions $(e^{\lambda(T-t)} w(t))^{2/3}$ and $w(t)^{-2/3}$ and exponents $p = 3/2, q = 3$:

$$\begin{aligned} \int_0^T e^{2\lambda(T-t)/3} dt &= \int_0^T (e^{\lambda(T-t)} w(t))^{2/3} \cdot w(t)^{-2/3} dt \\ &\leq \left(\int_0^T e^{\lambda(T-t)} w(t) dt \right)^{2/3} \left(\int_0^T \frac{1}{w^2(t)} dt \right)^{1/3} \leq B^{1/3} \left(\int_0^T e^{\lambda(T-t)} w(t) dt \right)^{2/3} \end{aligned} \tag{16}$$

It follows that $\int_0^T e^{\lambda(T-t)} w(t) dt \geq \frac{1}{B^{1/2}} \left(\frac{3(e^{2\lambda T/3} - 1)}{2\lambda} \right)^{3/2}$. Equality occurs when $e^{\lambda(T-t)} w(t)$ is proportional to $\frac{1}{w^2(t)}$ which means that $w(t)$ is proportional to $e^{-\lambda(T-t)/3}$.

Remark 1. The proof technique here works also for time-dependent λ .

2.3 Optimal time sampling distribution

The proposition 1 states that, at resources level B one cannot do better than (14). The question is how can one choose the right ρ in order to reach this minimal error level. The answer is in the next result.

Proposition 2. *Under the hypothesis (H Opt) the minimization problem corresponding to the loss L_ρ in (10) and the computational constraint (11) is guaranteed to obtain the best error level of proposition 1 only when ρ is the density of the exponential truncated law $\mathcal{E}^{0,T,4\lambda/3}$.*

Proof : Let us write

$$\begin{aligned} \left(\int_0^T \rho(t)^{1/2} \right) &\leq \left(\int_0^T \frac{1}{w^2(t)} dt \right)^2 \left(\int_0^T w^2(t) \rho(t) dt \right)^2 \\ &\leq \left(\int_0^T \frac{1}{w^2(t)} dt \right)^2 \left(\int_0^T w^2(t) \rho(t) dt \right)^2 \leq B^2 \left(\int_0^T w^2(t) \rho(t) dt \right)^2 = B^2 L_\rho^2. \end{aligned} \tag{17}$$

The loss will be minimized when there is equality in the above inequality which means that $1/w^2$ is proportional to $w^2 \rho$ that is ρ is proportional to w^{-4} . On the other hand, given proposition 1, the proof of proposition 2 is a matter of asking for which ρ the minimizer $w(s)$ of L_ρ under constraint (11) will be proportional to $e^{-\lambda(T-t)/3}$. Putting together these two arguments we obtain that overall error loss $|\mathcal{U}_\theta(T) - u(T)|$ is minimized when ρ is proportional to $e^{4\lambda(T-t)/3}$ i.e., it corresponds to the truncated law $\mathcal{E}^{0,T,4\lambda/3}$.

Remark 2. So we proved that under hypothesis (H Opt) concerning the algorithm’s convergence speed the error is minimal when the time sampling follows a truncated exponential law. The same result holds true if instead we consider algorithms with exponential convergence (12), see remark in the beginning of the proof of proposition 1.

2.4 Remarks on general settings: regimes of Lyapunov exponents

The example (7) may seem somehow too simple but in fact covers many situations encountered in practice. To this end let us recall the concept of *maximal Lyapunov exponent* used in the study of dynamical systems, particularly in chaos theory, to characterize the behavior of trajectories. The maximal Lyapunov exponent quantifies the rate of exponential divergence or convergence of nearby trajectories in the system.

Consider a dynamical system described by ordinary differential equation $u'(t) = \mathfrak{F}(t, u(t))$. If one considers two similar initial conditions $u(0)$ and $u(0) + \delta u(0)$ with $\delta u(0)$ playing the role of a small perturbation, the distance between these trajectories evolves over time according to the linearized dynamics around the system's trajectory. Specifically, if $\delta u(t)$ represents the perturbation at time t , then $\delta u(t)' = \nabla_u \mathfrak{F}(t, u(t)) \delta u(t)$. The maximal Lyapunov exponent λ (see [3, section 5.3]) is defined (for dimension 1) as the average exponential rate of separation of those trajectories: $\lambda = \lim_{t \rightarrow \infty} \lim_{\|\delta u(0)\| \rightarrow 0} \frac{1}{t} \ln \left(\frac{\|\delta u(t)\|}{\|\delta u(0)\|} \right)$. This can be also read as $\|\delta u(t)\| \sim e^{\lambda t} \|\delta u(0)\|$. The maximal Lyapunov exponent characterizes the system's sensitivity to initial conditions and can provide insights into whether the system exhibits chaotic behavior. In particular :

1. if $\lambda > 0$ close trajectories diverge exponentially, indicating chaotic behavior; in this case truncated exponential time sampling (weighting) is mandatory as it will be seen in numerical examples ;
2. if $\lambda < 0$ close trajectories converge exponentially, suggesting stability. In this case no special time sampling or weighting seems necessary and one may even imagine that an inverse sampling that puts more weight on larger time values can do better because initial perturbations fade away exponentially fast. This would correspond to parabolic evolution like the heat equation. In this situation the system evolves towards a static equilibrium.
3. if $\lambda = 0$ trajectories neither converge nor diverge, indicating a marginally stable or periodic behavior. Here for safety some truncated exponential sampling can be enforced. Systems involving conservation laws (not evolving towards static equilibrium) are present in this case (for PDE this will be called hyperbolic equations).

Note that if one is interested only in what happens in a neighborhood of the initial point and for small times, the corresponding concept is the *local Lyapunov exponent* which is related to the spectrum of the Jacobian $\nabla_u \mathfrak{F}$; note that this local metric can change regime as happens for instance with the Lorentz system.

3 Numerical examples

All these numerical tests are reproducible using codes available on Github ¹.

¹ https://github.com/gabriel-turinici/pinn_exponential_sampling version August 31st 2024.

3.1 Example in section 2

We investigate first the example in section 2 and describe below the numerical parameters and setting.

NN architecture and training parameters A neural network is used that will construct the mapping from t, x to the numerical solution $\mathcal{U}_\theta(t)$ as in figure 1. When the spatial dimension is not present, as is the case in section 2, then $\mathcal{U}_\theta(t)$ has a single input which is the time t . The NN has 5 fully connected (FC) layers with Glorot uniform initialization seeded with some constant for reproducibility; we checked that any other seed gives similar results. Each layer has 10 neurons and 'tanh' activation. This activation is classical in PINN because ReLU would give null second order derivatives. Then a final FC layer with no activation and one final neuron is used to output the model prediction.

We set $T = 1$. The initial value u_0 is taken to some non-special value i.e. not 0 or 1; here $u_0 = \sqrt{15}$ but many other values have been tested and give similar results. The shifting trick (9) in section 2 is used to be sure that the initial condition is not an issue and will be respected exactly (otherwise one has to study also the impact of the regularization coefficient used to impose the initial condition). The loss function (10) is employed and the sampling is performed with a truncated exponential law of rate r which is not necessarily equal to λ (recall that in general λ is unknown). The loss is computed by taking 100 sampling points which are exactly the quantiles of the law $\mathcal{E}^{0,T,r}$, see formula (22). We take $N_{iter} = 500$ iterations and an Adam optimizer with default learning rate (in our TensorFlow 2.15.0 version this is 10^{-3}). We checked that the quality of the numerical results can be made better by taking more iterations.

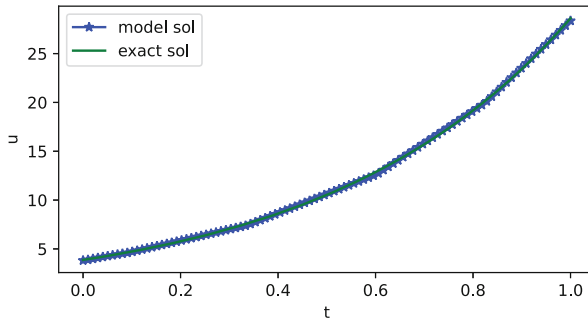


Fig. 2. Test of the model expressiveness (results for $\lambda = 2$). The model solution is graphically indistinguishable from the exact solution meaning that the NN is complex enough to reproduce the shape of the solution.

Validation of the NN architecture : expressiveness We checked first that the model is expressive enough. This means that, without any PINN framework, we just checked that the NN architecture can produce functions close enough to the exact solution $u(t)$. We used Adam optimizer with default parameters and mean square error between the model and the known exact solution. The results are plotted in figure 2 and show that the model is indeed expressive enough. Of course, this is just a theoretical possibility as the exact solution is in practice unknown and has to be found through the minimization of the PINN loss functional. But it still says that a good design of the loss functional should give good numerical results, i.e. that the model architecture is not the limiting hyper-parameter.

Validation of the PINN procedure: solution quality We now run the main PINN code. The solutions obtained for $\lambda = 2.0$ are plotted in figure 3. It is seen that the model is giving a good solution. This solution appears not enough converged for 500 epochs so we also gave the result for 1500 epochs where the numerical and exact solutions are indistinguishable graphically. This means that the PINN procedure is sound and gives expected results, in line with the literature.

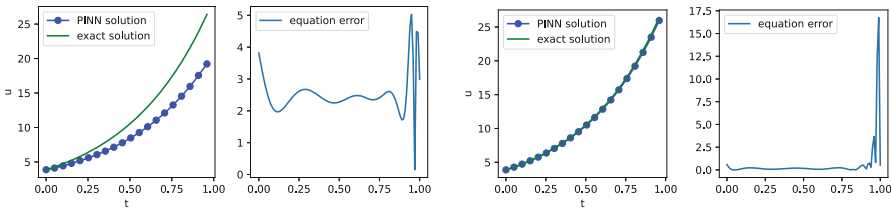


Fig. 3. Results for $\lambda = 2.$ and sampling parameter $r = 2.0$. First two plots: the results for 500 epochs. Last two plots: results for 1500 epochs.

Numerical results: solution sampling influence We move now to the main topic which is the influence of the sampling parameter r on the final error. In each case we set epoch number to 500 (similar results are obtained for any other number of epochs) to simulate a tight computational budget and

- set the λ parameter in a list enumerating all possible regimes: negative, null or positive, here $\lambda \in \{-2, 0, 2\}$;
- compute the performance for several sampling rate r parameters and look at the qualitative agreement with our theoretical results.

The numerical results are given in figure 4. Each point in the plot represents a NN trained from scratch with the PINN loss. For consistency of the comparison each NN starts from the same Glorot initialization with the same seed.

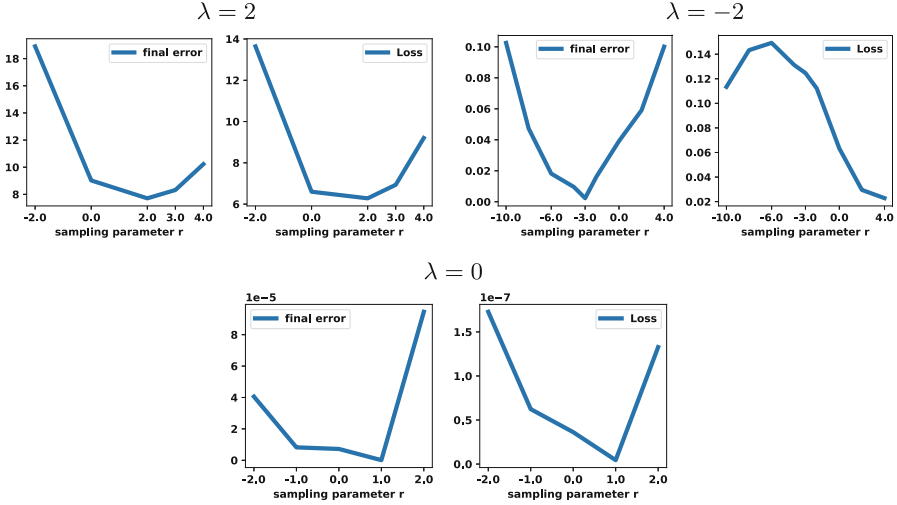


Fig. 4. Sampling law influence for $\lambda = 2$ (from left to right, top to bottom, plots 1 and 2), $\lambda = -2$ (plots 3 and 4) and $\lambda = 0$ (plots 5 and 6). All sampling are done with law $\mathcal{E}^{0,T,r}$. Plots 1, 3 and 5 : the final error as a function of r . Plots 2,4 and 6 : the loss.

Let start with plots 1 and 2 that correspond to $\lambda = 2$. It is seen that, among all possible truncated exponential laws, the one that gives minimal final error corresponds to positive value of r , which means that small t values are given more weight. This confirms the theoretical result in proposition 2 and is also intuitive because here $\lambda > 0$ which means exponential divergence of any perturbation. This exponential divergence has to be corrected by an effort to solve to higher precision the early dynamics. This is also coherent with the literature, see for instance [16] that discuss the importance of causality sampling. Note that in particular this empirical results confirms that uniform time sampling, which corresponds to $r = 0$ in the figure, is **not optimal**.

We move now to plots 3 and 4 (figure 4) that correspond to $\lambda = -2$. This dynamics converges to a stable equilibrium. In this case theory says that uniform sampling is not optimal and in fact giving **less** weight to initial times t is better because stability will erase most of the errors in this region. Note that this is **at odds** with previous results from the literature that encourage oversampling for low values of t irrespective of the regime. The numerical results confirm indeed that final error is minimized when sampling parameter r of the truncated exponential is negative (here optimal value is -3).

A special attention deserves the qualitative dependence of the loss on the sampling parameter r . Except for very negative r values, the loss decreases with r which would incorrectly suggest using large values of r . In fact here the loss is not informative; minimizing the loss is not the final goal, the final goal is to find the solution. The loss only encodes, as in reinforcement learning, the right information to find the solution. In this case, for negative values of r the loss

may appear larger but this happens because it works harder towards improving the outcome which is the final error. Therefore comparing two loss functionals corresponding to two different time sampling parameters r will not give the expected intuitive results **and will mislead the experimenter**.

Finally, the plots 5 and 6 (figure 4) correspond to $\lambda = 0$. Here the final error seems to be low over a plateau of parameters r around the value $r = 0$ (optimal appears to be reached for $r = 1$). So for this case the precise sampling parameter has less influence as long as it results in a somehow uniform time sampling. This is consistent with intuitive results and our theoretical results but again not always mentioned in the literature.

3.2 Burgers' equation

A test case often encountered in PINN applications is the Burgers' equation that describes the evolution of a one-dimensional viscous fluid flow. This nonlinear partial differential equation reads :

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad u(0, x) = -\sin(\pi x), \quad u(\pm 1, t) = 0 \quad \forall t \in [0, 1], \quad (18)$$

where $u = u(x, t)$ is the velocity field, $x \in [-1, 1]$ is the spatial variable and $\nu = 0.01/\pi$ is the viscosity coefficient.

Here, we consider as in [11] a neural network consisting of 9 fully connected 20-neurons layers with 'tanh' activation. We take a space-time grid with 25 spatial points and 50 time quantiles (see previous section). We also use the trick in (9) to impose exact initial condition.

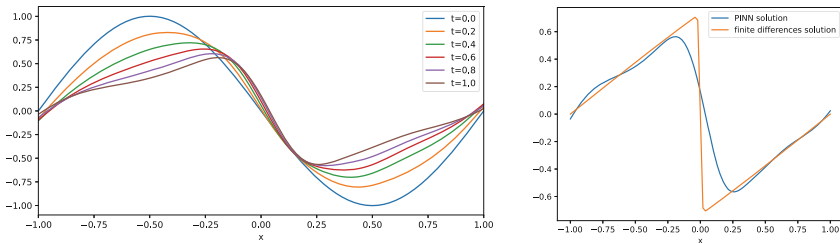


Fig. 5. Burgers' equation sampling parameter $r = 0$ i.e., uniform law $\mathcal{E}^{0,T,0}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

The results are given in figures 5 and 6. It is seen that both laws give similar results and in practice the computation of the norm of the difference at the final time indicates that the uniform sampling is better. To explain this result we need to recall that, even if the Burgers' equation is nonlinear and one could expect to find chaotic behavior similar to turbulence, however, the Hopf-Cole transformation allows to see that there is no substantial sensitivity with respect to initial

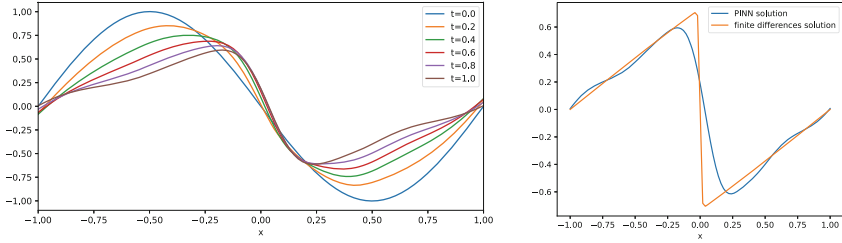


Fig. 6. Burgers' equation sampling parameter $r = 1$ and law $\mathcal{E}^{0,T,r}$. Left plot: the solution at different times. Right plot: the comparison with a finite difference solution considered exact.

conditions; in fact this equation is transformed to a linear parabolic equation. So the optimal sampling has no reason to overweight initial time instants and this is what we see here.

3.3 Lorenz system

For the final results we move to the Lorenz system, known to be chaotic, that has already been studied in the framework of PINN [16] :

$$x'(t) = \sigma(y - x), \quad y'(t) = x(\rho - z) - y, \quad z'(t) = xy - \beta z. \quad (19)$$

Here x, y, z are the state variables and σ, ρ, β are parameters : σ is the Prandtl number, ρ is the Rayleigh number, β is a parameter related to the aspect ratio of the system. We take as in [16] $\sigma = 10, \rho = 28, \beta = 8/3$ and initial state $(1, 1, 1)$.

To be coherent with previous implementations, instead of sampling under the law $\mathcal{E}^{0,T,r}$ we take uniform sampling but use the density of $\mathcal{E}^{0,T,r}$ as weight i.e. use time weighting proportional to e^{-rt} . We use the same NN as in section 3.1 but with 20 neurons per layer, 10'000 iterations and shifting trick in (9). The results are given in figure 7. In this case the uniform i.e., $r = 0$ weighting does not manage to obtain a reasonable solution. This is understood from the fact that the equation error remains large at the initial times and, due to the chaotic behavior of the system, divergence with respect to the correct trajectory will occur. The same happens for $r = -10$ which over-weights the final time instants as the expense of the initial ones. On the contrary, putting more weight on the initial time instants as is done in figure 7 (bottom three plots) will bring the numerical solution very close to the exact solution.

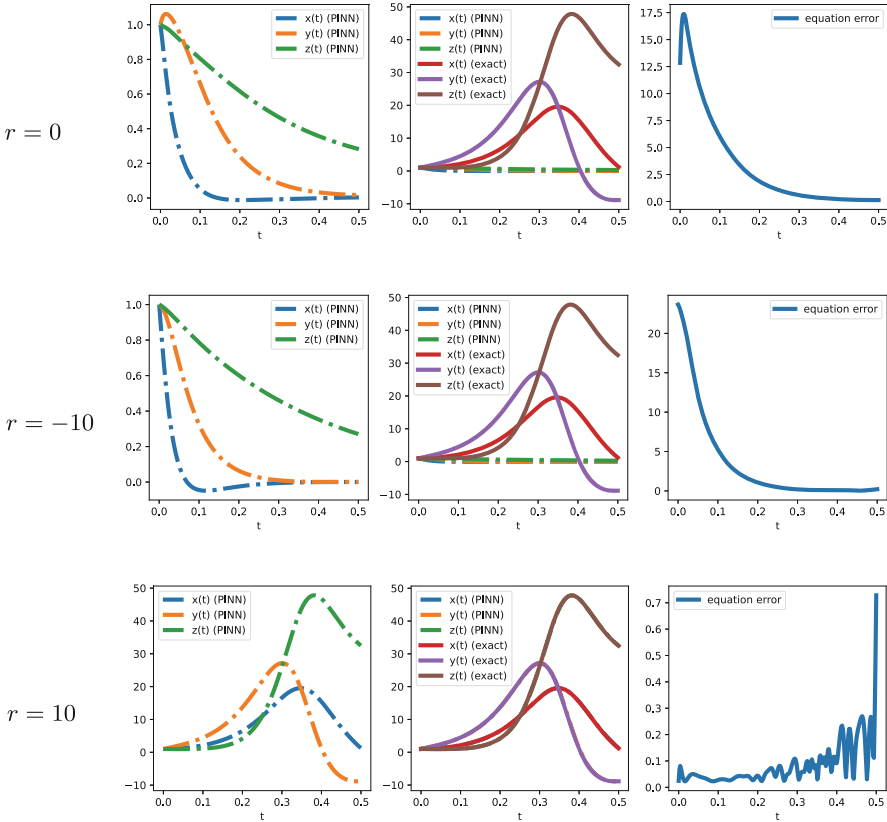


Fig. 7. Lorenz system with weight parameters $r = 0$ (first row of plots), $r = -10$ (second row of plots) and $r = 10$ (third row of plots). The approximation quality is not good for $r = 0$ and $r = -10$ and much improved for $r = 10$, in fact the numerical and exact solutions are superposed and indistinguishable graphically.

4 Discussion and conclusion

The goal of the PINN framework is to find the solution to a given evolution equation. This goal is transcribed through the use of a loss functional. Many loss functionals give, in the limit of infinite computational budget, the same optimal solution. But in practice the computational budget is limited and not all loss functionals behave alike. We discuss here the effect of the temporal sampling on the error of the solution at the final time; to this end we prove for the first time that, under hypothesis regarding the optimization algorithm, the optimal sampling belongs to the class of truncated exponential distribution. We additionally characterize the optimal distribution parameter. The qualitative insight is that when the evolution is chaotic or sensitive to initial conditions early time instants should be given more weight (exponentially). On the contrary when the

evolution is periodic or stably converging to an equilibrium this over-weighting is not useful any more. The theoretical results were checked numerically on several important examples and the empirical observations are coherent with them.

The principal limitation of the work is that the optimal sampling parameter is in general unknown and has to be selected in the usual manner of hyper-parameter search. Future work will hopefully shed some light on what is the best practice to reach this optimal sampling regime.

A Appendix : truncated exponential distribution

A truncated exponential distribution is described by a triplet of parameters (a, b, r) , $a \leq b$; the parameters a and b define the support of the distribution $[a, b] \subset \mathbb{R}$ while the rate r defines the speed of decay. The distribution is by definition the only probability measure $\mathcal{E}^{a,b,r}$ with support in $[a, b]$ and density proportional to e^{-rt} , i.e.,

$$\mathcal{E}^{a,b,r}(dt) = r \frac{e^{-rt}}{e^{-ra} - e^{-rb}} \mathbb{1}_{[a,b]} dt. \tag{20}$$

Note that it is not required that $r \geq 0$. When $r \rightarrow 0$ we obtain the uniform distribution on $[a, b]$ denoted $U(a, b)$. When $a = 0, b = \infty$ we obtain the (non-truncated) exponential distribution of rate r . To sample from this law, direct computations allow to show that ²:

$$\text{If } U \sim U(0, 1) \text{ then } Y = \frac{-\ln(1 - U + Ue^{-r(b-a)})}{r} \sim \mathcal{E}^{a,b,r}. \tag{21}$$

In particular the q -quantile of this distribution is precisely

$$\frac{-\ln(1 - q + qe^{-r(b-a)})}{r}. \tag{22}$$

B Appendix : further quality metrics

One could ask what happens when our main output is not the solution at final time T but some integral over all times, i.e., instead of $\delta u(T)$ our quality metric is :

$$\int_0^T |\delta u(t)| dt = \int_0^T |\mathcal{U}_\theta(t) - u(t)| dt. \tag{23}$$

This is answered in the following result.

Proposition 3. *Under the hypothesis (H Opt) the minimization problem corresponding to the loss L_ρ in (10) and the computational constraint (11) is guaranteed to obtain the best error level for the metric (23) only when*

$$\rho(t) = \frac{(e^{-\lambda t} - e^{-\lambda T})^{4/3}}{\int_0^T (e^{-\lambda t} - e^{-\lambda T})^{4/3}}. \tag{24}$$

² Here $X \sim \mu$ means that the random variable X follows the law μ .

Note that although the density in (24) is not exactly a truncated exponential, it will become one in the limit $T \rightarrow \infty$.

Proof : Many parts of the proofs are the same as soon as we recognize that, under same hypothesis, the formula of the error metric $\delta u(T)$ given in (15) can be replaced by

$$\int_0^T |\delta u(t)| dt = \int_0^T \int_0^t e^{\lambda(t-s)} w(s) ds dt = \int_0^T \frac{e^{\lambda(T-s)} - 1}{\lambda} w(s) ds. \quad (25)$$

The w that minimizes (25) under resources constraint (11) is found as before to be proportional to $(e^{\lambda(T-s)} - 1)^{1/3}$. The rest follows as before.

References

1. Ștefana Anița, Turinici, G.: On the Convergence Rate of the Stochastic Gradient Descent (SGD) and Application to a Modified Policy Gradient for the Multi Armed Bandit (2024), [arxiv:2402.06388](https://arxiv.org/abs/2402.06388)
2. Bae, H.O., Kang, S., Lee, M.: Option Pricing and Local Volatility Surface by Physics-Informed Neural Network. *Comput. Econ.* (2024). <https://doi.org/10.1007/s10614-024-10551-2>
3. Cencini, M., Cecconi, F., Vulpiani, A.: *Chaos*. WORLD SCIENTIFIC (2009). <https://doi.org/10.1142/7351>, <https://worldscientific.com/doi/abs/10.1142/7351>
4. Chen, H.F.: *Stochastic approximation and its applications*, Nonconvex Optim. Appl., vol. 64. Dordrecht: Kluwer Academic Publishers (2002)
5. Fehrman, B., Gess, B., Jentzen, A.: Convergence rates for the stochastic gradient descent method for non-convex objective functions. *Journal of Machine Learning Research* **21**(136), 1–48 (2020), <http://jmlr.org/papers/v21/19-636.html>
6. Hu, Z., Shukla, K., Karniadakis, G.E., Kawaguchi, K.: Tackling the curse of dimensionality with physics-informed neural networks (2024)
7. Liu, S., Chen, X., Di, X.: Scalable learning for spatiotemporal mean field games using physics-informed neural operator. *Mathematics* **12**(6) (2024). <https://doi.org/10.3390/math12060803>, <https://www.mdpi.com/2227-7390/12/6/803>
8. Mertikopoulos, P., Hallak, N., Kavis, A., Cevher, V.: On the Almost Sure Convergence of Stochastic Gradient Descent in Non-Convex Problems. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1117–1128. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/0cb5ebbb1b34ec343dfe135db691e4a85-Paper.pdf, [arxiv:2006.11144](https://arxiv.org/abs/2006.11144)
9. Penwarden, M., Jagtap, A.D., Zhe, S., Karniadakis, G.E., Kirby, R.M.: A unified scalable framework for causal sweeping strategies for Physics-Informed Neural Networks (PINNs) and their temporal decompositions. *Journal of Computational Physics* **493**, 112464 (2023) <https://doi.org/10.1016/j.jcp.2023.112464>, <https://www.sciencedirect.com/science/article/pii/S0021999123005594>
10. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes. The art of scientific computing*. Cambridge: Cambridge University Press, 3rd ed. edn. (2007)

11. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving non-linear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019). <https://doi.org/10.1016/j.jcp.2018.10.045>, <https://www.sciencedirect.com/science/article/pii/S0021999118307125>
12. Robbins, H., Monro, S.: A Stochastic Approximation Method. *The Annals of Mathematical Statistics* **22**(3), 400 – 407 (1951). <https://doi.org/10.1214/aoms/1177729586>, <https://doi.org/10.1214/aoms/1177729586>, publisher: Institute of Mathematical Statistics
13. Soohan, K., Yun, S.B., Hyeong-Ohk, B., Muhyun, L., Youngjoon, H.: Physics-informed convolutional transformer for predicting volatility surface. *Quantitative Finance* **24**(2), 203–220 (2024). <https://doi.org/10.1080/14697688.2023.2294799>
14. Subramanian, S., Kirby, R.M., Mahoney, M.W., Gholami, A.: Adaptive self-supervision algorithms for physics-informed neural networks. arXiv preprint [arXiv:2207.04084](https://arxiv.org/abs/2207.04084) (2022), eCAI 2023 Proceedings
15. Turinici, G.: The convergence of the Stochastic Gradient Descent (SGD) : a self-contained proof (2023). <https://doi.org/10.5281/ZENODO.4638694>, [arxiv:2103.14350v2](https://arxiv.org/abs/2103.14350v2)
16. Wang, S., Sankaran, S., Perdikaris, P.: Respecting causality is all you need for training physics-informed neural networks. arXiv preprint [arXiv:2203.07404](https://arxiv.org/abs/2203.07404) (2022)
17. Wang, S., Teng, Y., Perdikaris, P.: Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing* **43**(5), A3055–A3081 (2021). <https://doi.org/10.1137/20M1318043>, <https://doi.org/10.1137/20M1318043>
18. Wight, C.L., Zhao, J.: Solving Allen-Cahn and Cahn-Hilliard equations using the adaptive physics informed neural networks. arXiv preprint [arXiv:2007.04542](https://arxiv.org/abs/2007.04542) (2020)



HeFormer: A Lightweight Transformer Combining Hash Estimation for Link Prediction

Teng Sun^{1,2}(✉), Xiaoqiang Xiao^{1,2}, Xu Zhang¹, and Weixun Ning¹

¹ School of Computer, National University of Defense Technology, Changsha, China
{sunteng,xqxiao,zhangxu09a,ningweixun}@nudt.edu.cn

² State Key Laboratory of High Performance Computing, Changsha, China

Abstract. Link Prediction(LP) is a fundamental problem in graph machine learning that aims to predict the existence of links between nodes. Most current research on LP adopts Graph Neural Networks (GNNs) to learn the representation of subgraphs, but fails to efficiently capture global topological information in large graphs. In response to this issue, we focus on the global attention mechanism of Transformers. Nevertheless, original Transformers are not inherently suitable for learning graph-structured data, and their deep multi-head attention architecture has been limited by prohibitive compute and memory costs. In this work, we propose a lightweight model: a single-layer, single-head Transformer, which constructs subgraph structural features based on hash estimation. It provides a new perspective for applying Transformers in graph-structured data processing. Firstly, we utilize MinHash and HyperLogLog techniques to estimate the structural information of subgraphs, then fuse subgraph structural features with node features to achieve efficient message passing. In this case, our model does not need to extract or manipulate enclosed subgraphs, and structural features can be pre-processed. Additionally, we design a single-layer, single-head Transformer as the encoder for graph-structured data, utilizing the attention mechanism to capture global effects. Meanwhile, our model does not require extra positional encoding, which significantly reduce computational complexity. Extensive experiments demonstrate that our model achieves optimal prediction accuracy on large-scale datasets with high efficiency. The code is available at <https://github.com/sunteng6/HeFormer>.

Keywords: Hash estimation · Transformers · Feature learning · Link prediction · Graph machine learning

1 Introduction

In the real world, complex systems are typically described as networks composed of nodes and edges (e.g., biological networks, social networks, citation networks, transportation networks). Nodes represent various entities in complex systems,

while edges describe the relationships between these entities[1].;Link prediction enhances the understanding of the relationships between specific node pairs and the overall evolution of the network, and has widespread applications in real-world scenarios. In biology, LP is used to predict unobserved interactions in protein-protein interaction (PPI) networks, thereby facilitating new drug development and advancing biological research[2]. In the analysis of disease transmissibility, such as COVID-19, LP infers possible interactions to help with tracing disease transmission pathways[3]. In social networks, LP assist users in discovering friends with similar interests or recommending appealing products.

Current approaches have adapted GNNs for learning representations over graph-structured data, achieving strong performance in link prediction tasks[4]. However, most GNN-based methods require extracting and embedding subgraphs for each pair of nodes, leading to high computational complexity. Additionally, multiple rounds of GNN message passing can result in overly similar feature representations, leading to information loss and over-smoothing. To address the issue of long-range dependencies in GNNs, Transformers have emerged as foundational encoders for graph-structured data[5]. Their self-attention mechanism captures these dependencies by aggregating all node embeddings to update each node’s representation[6]. Nevertheless, Transformers are not inherently suitable for graph-structured data and require additional design to leverage the graph’s topological information. Moreover, the computational complexity of deep multi-head attention architecture is $O(N^2)$ [7]. In large-scale datasets, the computational challenges of processing subgraphs and employing a global attention mechanism become particularly evident, as both computation time and storage overhead increase significantly.

To overcome these inefficiencies, we propose **HeFormer**, a novel model that combines **H**ash estimation with a lightweight **Tran**s**Form**er. The overall architecture is illustrated in Fig. 1. This model consists of four main components: (1) graph-structured data input module; (2) feature generator that constructs structural features features using Minihash and HyperLogLog hash estimation techniques, and preprocesses features; (3) single-layer, single-head Transformer encoder that simplifies computation based on kernel function approximation; (4) link prediction evaluation module. Our technical contributions are detailed below:

- We leverage the global attention of Transformers to learn hash-estimated structural features, which can reduce information loss compared to GNN-based link prediction methods. Our model, HeFormer, constructs and preprocesses structural features ahead of message passing, eliminating the need to extract and manipulate subgraphs for each pair of nodes.
- We design a lightweight, single-layer, single-head Transformer as the encoder for graph-structured data. Our model significantly alleviates both time and memory overhead. By discarding exponential terms and softmax operations, we alter the matrix computation order of original Transformers[8], reducing the encoder’s complexity from $O(N^2d)$ to $O(Nd^2)$.

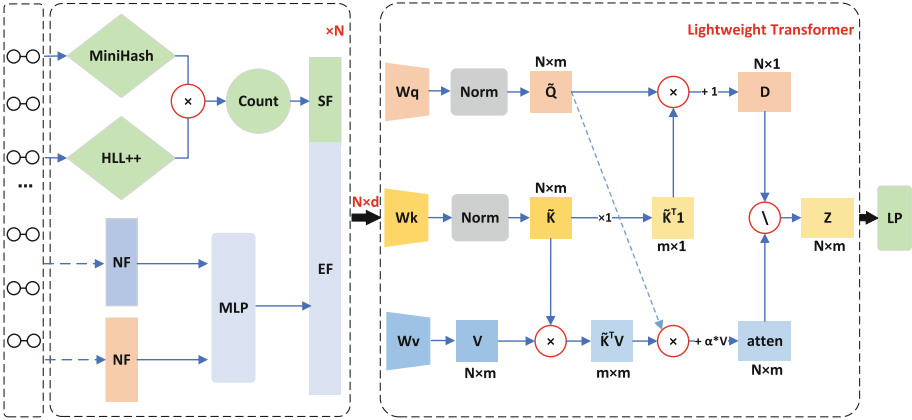


Fig. 1. Illustration of our proposed HeFormer and its data flow. HeFormer combines counts from different hop distances to construct structural features (SF), while node features (NF) are processed through the Hadamard product and an MLP to generate edge features (EF). SF and EF are then concatenated and fed into a single-layer, single-head Transformer for message passing. Finally, the link prediction (LP) performance is evaluated.

- We compare the performance of HeFormer with mainstream link prediction methods on real-world datasets. Experimental results demonstrate that HeFormer achieves optimal prediction accuracy and lower computational costs on large-scale datasets. Feature ablation and sensitivity analyses confirm the effectiveness of our modules.

2 Related Works

2.1 Link prediction

Current mainstream link prediction methods leverage Graph Neural Networks to aggregate features from nodes and their neighbors to learn node and link representations[3]. The classic algorithm SEAL[9] employs GNNs to reformulate the link prediction task into a binary classification problem focused on subgraphs. However, SEAL requires extracting subgraphs around each target link, which is computationally expensive when dealing with large graphs. To mitigate the substantial overhead associated with explicitly constructing subgraphs, ELPH[10] leverages hashing techniques to craft subgraph structural features and transmits subgraph sketches as messages. Nonetheless, ELPH is still a GCN-based model, which encounters GPU memory constraints when managing large-scale datasets. Based on ELPH, BUDDY[10] is a scalable model that pre-computes sketches and node features to improve efficiency. However, by replacing GCN with MLP, BUDDY is unable to directly leverage the neighborhood information of nodes, resulting in a lack of interpretability when capturing complex graph structural

information. By contrast, our model utilizes the global attention mechanism of Transformer to enhance feature learning capabilities.

2.2 Hash estimation techniques

As the foundation for constructing structural features, we concurrently employ two sketching techniques to estimate the counts of node neighborhood intersections. MinHash[11] is an algorithm to estimate the Jaccard similarity between two sets: $Jaccard(A, B) = |A \cap B| / |A \cup B|$. The Jaccard similarity is defined as the ratio of the intersection to the union, serving as a measure of sets similarity.

HyperLogLog[12] is an efficient algorithm designed to estimate the cardinality of large sets (i.e., the number of distinct elements). It operates by mapping elements to binary strings using a hash function, then calculating the length of the leading zero prefix in each binary string. Elements are assigned to different buckets based on the first few bits of their hash values. Each bucket records the maximum zero prefix length, and the harmonic mean of these lengths is calculated. Finally, by applying a correction factor, the estimated cardinality of the set is obtained. MinHash and HyperLogLog are memory-efficient and fast, making them suitable for handling large-scale datasets.

2.3 Graph Transformers

Transformers have the ability to capture implicit dependencies beyond neighboring nodes, and recent research has sought to extend the original Transformers architecture to graph-structured data. However, the time and space complexity of Transformer typically increases exponentially with the number of nodes. As a result, training deep Transformer on large graphs with hundreds of thousands of nodes is extremely resource-intensive. To address this issue, NodeFormer [13] introduced a kernelized Gumbel-Softmax operation, reducing the algorithmic complexity of message passing to linear growth with the number of nodes. Although this method maintains accuracy in node classification tasks, it still employs a deep multi-head attention mechanism. Building on the concept of NodeFormer, SGFormer[5] proposed a simple single-layer Transformer architecture for node classification tasks. However, it still needs to be combined with GCN, increasing the model’s complexity and failing to fully overcome the limitations of GNNs.

3 Proposed Method

In light of these issues, we design a lightweight and compute efficient link prediction model called HeFormer, which leverages hash estimation techniques (Mini-Hash and HyperLogLog) to compute set cardinality and construct structural features (Sect. 3.1). Then, we provide an intuitive theoretical analysis of why our single-layer, single-head Transformer model is lightweight (Sect. 3.2).

3.1 Hash estimation for constructing structural features

In distance encoding (DE), each node is encoded with a tuple $z_{ij} = (d(u, i), d(v, j))$ [14], as illustrated in Fig. 2(b). By fixing the number of DE labels (setting the maximum distance to three), we directly construct structural features by combining the counts of intersections of neighborhood nodes up to $h(h + 2)$ dimensions, where h is the maximum hop count of 3.

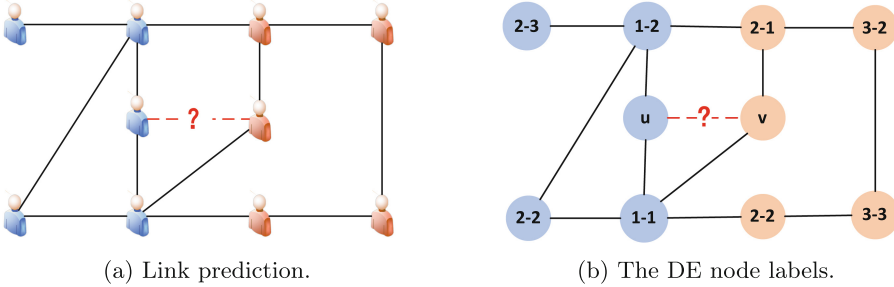


Fig. 2. The number of nodes at distances d_u and d_v from u and v represent the structural information. (a) Link prediction infers the likelihood of a connection between two users based on the structure of the social network. (b) The distance encoding (DE) labels of the neighbor node for link (u, v) .

MiniHash approximates the Jaccard similarity by generating the minimum hash values using multiple different hash functions. For example, for a link (u, v) , k different hash functions h_1, h_2, \dots, h_k are randomly initialized for nodes u and v , resulting in their MiniHash signatures $MH(u)$ and $MH(v)$ respectively:

$$\begin{aligned}
 MH(u) &= [h_1(u), h_2(u), \dots, h_k(u)] \\
 MH(v) &= [h_1(v), h_2(v), \dots, h_k(v)]
 \end{aligned}
 \tag{1}$$

For each hash function, find the minimum hash value among all elements in the set. The collection of these minimum hash values is called the MinHash signature. By comparing the MinHash signatures of two sets using Eq. (2), we calculate the proportion of positions in the signatures where the minimum hash values are equal. This proportion serves as the estimated Jaccard similarity.

$$|MH(u) \cap MH(v)| = \sum_{i=1}^k \mathbb{I}(h_i(u) = h_i(v))
 \tag{2}$$

where \mathbb{I} is an indicator function that equals 1 when $h_i(A) = h_i(B)$. According to the graph structure’s connectivity, the hash functions are propagated. The hash value update strategy for nodes u and v follows Eq. (3). As the number of hops

increases, each node's k hash values are updated based on its neighborhood:

$$\begin{aligned} MH(u)^{(i)} &= \min_{v \in \mathcal{N}(u)} MH(v)^{(i-1)} \\ MH(v)^{(j)} &= \min_{u \in \mathcal{N}(v)} MH(u)^{(j-1)} \end{aligned} \quad (3)$$

where i, j denotes the number of hops, and \mathcal{N} represents the neighborhood of the node. The Jaccard index of the neighborhood sets of node u at hop i and node v at hop j can be approximated by the ratio of the number of equal hash values in their MiniHash signatures to the total number of hash functions:

$$\text{Jaccard}(u, v)^{(i, j)} \approx \frac{|MH(u)^{(i)} \cap MH(v)^{(j)}|}{k} \quad (4)$$

where k is the total number of hash functions. Similar to the MiniHash calculation steps, each node is initialized with p^2 HyperLogLog register values, which are then propagated according to the link relationships. The propagation process is described by Eq. (5). After (i, j) hops of propagating register values, the HLL hash values of the source node u and the target node v are first merged to estimate the count of the union of the neighborhoods of the two nodes. The merging operation of HLL involves taking the maximum value of the corresponding registers of the two nodes element-wise, ensuring that the merged hash value contains all elements from both the source and target nodes:

$$\begin{aligned} HLL(u)^{(i)} &= \max_{v \in \mathcal{N}(u)} HLL(v)^{(i-1)}, \\ HLL(v)^{(j)} &= \max_{u \in \mathcal{N}(v)} HLL(u)^{(j-1)} \end{aligned} \quad (5)$$

Hash functions uniformly distribute elements across the registers, with the number of leading zeros inversely related to the cardinality. By counting the number of leading zeros in the registers, we can approximate the number of distinct elements in the union of the neighborhoods of node u and its connected node v . It is worth noting that, to reduce computational cost, we set a threshold and initially use linear estimation to decrease the computational overhead:

$$\text{Count}_1(V_0) = m \times \log\left(\frac{m}{V_0}\right) \quad (6)$$

where m is the number of registers, $m = p^2$, and V_0 is the number of registers with a value of zero. if $\text{Count}_1 > \text{threshold value}$, the original HLL counting method is used:

$$\text{Count}_2(Z_i) = \alpha \cdot m^2 \cdot \left(\sum_{i=1}^m 2^{-Z_i}\right)^{-1} \quad (7)$$

where Z_i is the value of the i -th register and α is a tunable parameter. HyperLogLog can effectively estimate the cardinality of the union, as shown in Eq. (8), while MinHashing can estimate the *Jaccard* index. HyperLogLog and MinHashing can be combined to estimate the intersection and union of node sets:

$$\left|N(u)^{(i)} \cup N(v)^{(j)}\right| = \text{Count}\left(\max\left(HLL(u)^{(i)}, HLL(v)^{(j)}\right)\right) \quad (8)$$

where *max* refers to taking the maximum value of the *m* registers for the two nodes separately. The cardinality is calculated based on the values of all the bits in the registers. This combined method leverages the advantages of both HyperLogLog and MinHashing, enabling efficient set operations on large-scale graph data. Using MiniHash and HLL techniques, we can compute the neighborhood intersection counts between linked nodes at hops (1, 1), (1, 2), (2, 1), (1, 3), (3, 1), (2, 2), (2, 3), (3, 2), and (3, 3):

$$\begin{aligned}
 \text{Intersections}(u, v)^{(i,j)} &= |N_i(u) \cap N_j(v)| \\
 &= \text{Jaccard}(u, v)^{(i,j)} \cdot \left| N(u)^{(i)} \cup N(v)^{(j)} \right| \\
 &\approx \frac{|MH(u)^{(i)} \cap MH(v)^{(j)}|}{k} \cdot \text{Count} \left(\max \left(HLL(u)^{(i)}, HLL(v)^{(j)} \right) \right)
 \end{aligned}
 \tag{9}$$

We utilize these counts to construct structural features of the edges. For example, f_0 represents the intersection count of nodes u and v at hop (1, 1), f_2 represents the count at hop (1, 2) minus the count at hop (1, 1), and f_9 indicates the number of neighbors of node v at hop 1 but beyond the range of hop 3 for node u . This processing method captures detailed neighborhood information of the source and target nodes at different hops, including intersections and differences. The features $\{f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}\}$ are combined into a 15-dimensional feature matrix features, as illustrated in Fig. 3. The dimensions change with the number of hops. The generated structural features can represent the neighborhood information of each pair of nodes at different hops, and the

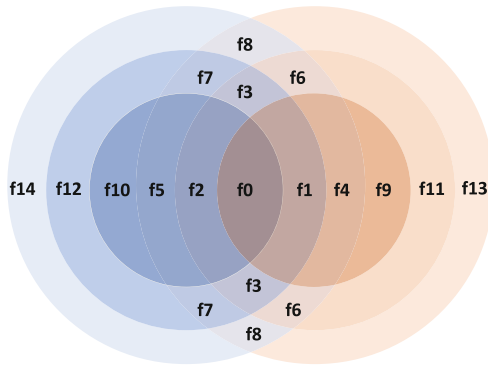


Fig. 3. Construction strategy for subgraph structural features. Flexibly combining based on the intersection numbers of the neighborhoods of node u and node v at 1, 2, and 3 hops.

dimensionality of these structural features is fixed, independent of the graph’s size. After integrating structural features and node features, this feature learning method can efficiently combine original features and graph structure without handling redundant subgraphs.

3.2 Lightweight single-layer, single-head Transformer

Link prediction methods based on GNNs typically rely on message passing between neighboring nodes. We aim to use the self-attention mechanism of Transformer to directly capture the global information in the graph. However, due to the need to stack deep multi-head attention mechanisms, the time and space complexity of original Transformers are $O(N^2)$. The computation process of the Query, Key, and Value (QKV) matrices is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (10)$$

As the number of training layers, the computational cost of processing graph data grows exponentially. Therefore, based on the architectures of NodeFormer [13] and SGFormer[5], we design a lightweight single-layer, single-head attention model, which achieves lower computational cost with linear complexity. Our model can efficiently propagate feature information across large graph data while ensuring the ability to learn link and node representations. The specific inference process is analyzed below. Firstly, from a matrix perspective, updating the representations of all edges:

$$\hat{\mathbf{A}}^{(l)} = \text{softmax}\left(\left(W_Q^{(l)}\mathbf{z}^{(l)}\right)^\top \left(W_K^{(l)}\mathbf{z}^{(l)}\right)\right), \quad \mathbf{z}^{(l+1)} = \hat{\mathbf{A}}^{(l)}W_V^{(l)}\mathbf{z}^{(l)} \quad (11)$$

where $W_Q^{(l)}$, $W_K^{(l)}$, and $W_V^{(l)}$ are the learnable parameters of the l -th layer, $\mathbf{z}^{(l)}$ is the vector representation. The Softmax calculation formula is given:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{i=1}^n e^{x_i}} \in (0, 1) \quad (12)$$

Therefore, we define a global attention network that transforms the Softmax operation into an exponential operation. This network estimates the potential interactions between instance nodes and implements the corresponding message passing. From the perspective of each node, it updates the feature propagation of each link u :

$$\mathbf{q}_u = W_Q^{(l)}\mathbf{z}_u^{(l)}, \quad \mathbf{k}_u = W_K^{(l)}\mathbf{z}_u^{(l)}, \quad \mathbf{v}_u = W_V^{(l)}\mathbf{z}_u^{(l)} \quad (13)$$

$$\mathbf{z}_u^{(l+1)} = \sum_{v=1}^N \frac{\exp(\mathbf{q}_u^\top \mathbf{k}_v)}{\sum_{w=1}^N \exp(\mathbf{q}_u^\top \mathbf{k}_w)} \cdot \mathbf{v}_v \quad (14)$$

where $\mathbf{z}_u^{(l)}$ represents the feature learned at the l -th layer. To reduce the complexity $O(N^2)$ of the above network and accelerate the global model, we can simplify the operation of taking the exponential of the dot product of transposed vectors using a positive definite kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ for pairwise similarity:

$$\mathbf{z}_u^{(l+1)} = \sum_{v=1}^N \frac{\kappa(\mathbf{q}_u, \mathbf{k}_v)}{\sum_{w=1}^N \kappa(\mathbf{q}_u, \mathbf{k}_w)} \cdot \mathbf{v}_v \quad (15)$$

A positive-definite kernel function can transfer the inner product computation from the original space to a high-dimensional space, thereby effectively handling nonlinear relationships. The kernel function can be further approximated by random features (RF)[15]. Here, Mercer’s theorem is invoked to represent the kernel function as an inner product in a high-dimensional space:

$$\kappa(\mathbf{a}, \mathbf{b}) = \langle \Phi(\mathbf{a}), \Phi(\mathbf{b}) \rangle \nu \approx \phi(\mathbf{a})^\top \phi(\mathbf{b}) \quad (16)$$

where $\Phi : \mathbb{R}^d \rightarrow V$ is a basis function that maps the input into a high-dimensional vector space V , while $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is a low-dimensional feature map that achieves an unbiased estimation through random transformation. By using a positive-definite kernel function and Mercer’s theorem[16], the exp operation in Eq. (14) is eliminated, altering the order of vector operations:

$$\mathbf{z}_u^{(l+1)} = \sum_{v=1}^N \frac{\phi(\mathbf{q}_u)^\top \phi(\mathbf{k}_v)}{\sum_{w=1}^N \phi(\mathbf{q}_u)^\top \phi(\mathbf{k}_w)} \cdot \mathbf{v}_v = \frac{\phi(\mathbf{q}_u)^\top \sum_{v=1}^N \phi(\mathbf{k}_v) \cdot \mathbf{v}_v^\top}{\phi(\mathbf{q}_u)^\top \sum_{w=1}^N \phi(\mathbf{k}_w)} \quad (17)$$

Moreover, the double summation in the numerator and denominator is shared by each vector representation u , thus only needing to be computed once. This reduces the computational complexity of feature message passing across the entire graph from $O(N^2)$ to $O(N)$. In practical applications, instead of propagating updates for individual edges, matrix operations are used:

$$\mathbf{D}^{(l)} = \left[\tilde{\mathbf{Q}}^{(l)} \left((\tilde{\mathbf{K}}^{(l)})^\top \mathbf{1} \right) \right]^{-1} \quad (18)$$

$$\mathbf{Z}^{(l+1)} = \mathbf{D}^{(l)} \left[\tilde{\mathbf{Q}}^{(l)} \left((\tilde{\mathbf{K}}^{(l)})^\top \mathbf{V}^{(l)} \right) \right] \quad (19)$$

where $\mathbf{Q}^{(l)} = W_Q \mathbf{Z}^{(l)}$, $\mathbf{K}^{(l)} = W_K \mathbf{Z}^{(l)}$, $\mathbf{V}^{(l)} = W_V \mathbf{Z}^{(l)}$, $\tilde{\mathbf{Q}}^{(l)} = \phi(\mathbf{Q}^{(l)})$, $\tilde{\mathbf{K}}^{(l)} = \phi(\mathbf{K}^{(l)})$. The aforementioned network propagation still tends to stack deep multi-head attention layers to obtain effective feature representations. To further simplify the computation and achieve a single-layer, single-head attention propagation mechanism, we use the following defined linear attention function:

$$\tilde{\mathbf{Q}} = \frac{W_Q \mathbf{Z}^{(0)}}{\|W_Q \mathbf{Z}^{(0)}\|_{\mathcal{F}}}, \quad \tilde{\mathbf{K}} = \frac{W_K \mathbf{Z}^{(0)}}{\|W_K \mathbf{Z}^{(0)}\|_{\mathcal{F}}}, \quad \mathbf{V} = W_V \mathbf{Z}^{(0)} \quad (20)$$

where W_Q , W_K , and W_V are linear feed-forward layers. $\|\cdot\|$ denotes the Frobenius norm. Then, we employ addition and the parameter α to retain the information of the central node while considering the input graph structure information:

$$\mathbf{D} = \left[\mathbf{1} + \tilde{\mathbf{Q}} \left(\tilde{\mathbf{K}}^\top \mathbf{1} \right) \right]^{-1}, \quad \mathbf{Z} = \mathbf{D} \left[\alpha \mathbf{V} + \tilde{\mathbf{Q}} \left(\tilde{\mathbf{K}}^\top \mathbf{V} \right) \right] \quad (21)$$

where $\mathbf{Z}^{(0)}$ is the original input vector representation. In our model, it represents the link information obtained by concatenating node and structural features. And $\mathbf{1}$ is an N -dimensional all-one column vector. The output \mathbf{Z} integrates

the pairwise attention propagation of N link representations. We visualized the matrix computation process of the proposed lightweight Transformer Fig. 4, where N denotes the number of link vector representations, d denotes the dimension and $l = \frac{d}{h}$. Compared to the original Transformers, it reduces the encoder complexity from $O(N^2d)$ to $O(Nd^2)$.

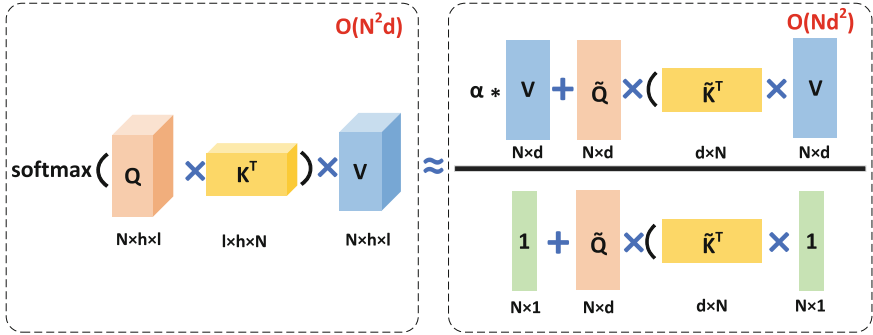


Fig. 4. (Left) Original Transformers computation process of QKV matrix vectors in the multi-head attention mechanism. **(Right)** HeFormer eliminates the softmax operation and changes the order of matrix multiplication. The addition operations allow the model to capture influences from other links while preserving information from the central link.

Link Prediction. Eq. (22) demonstrates how to apply the learned pair representations Z to the link prediction problem. The value of Y represents the likelihood of the existence of a link. We train our proposed model using the standard binary cross-entropy (*BCE*) loss:

$$Y = \sigma(f(\mathbf{Z})), \quad \mathcal{L} = \sum_{\hat{y}_{uv} \in Y} BCE(\hat{y}_{uv}, y_{uv}) \quad (22)$$

where f is a readout function, which in our work is a linear feed-forward layer. In the loss function, \hat{y}_{uv} is the probability value obtained through the sigmoid function $\sigma(\cdot)$, and y_{uv} is the actual label of link (u, v) , typically 0 or 1. Next, we will demonstrate through experiments that our lightweight single-layer, single-head Transformer not only has excellent feature learning capabilities but also scales effectively to large graphs.

4 Experiments

4.1 Experiment Settings

Datasets. We include the most widely used citation datasets for link prediction, such as Citeseer and Pubmed, as well as Open Graph Benchmark (OGB) datasets[17]: ogbl-collab and ogbl-ppa. The statistics of nodes and edges for each dataset are summarized in Table 1.

Table 1. The statistics of experimented datasets.

	Citeseer	Pubmed	Collab	PPA
#Nodes	3327	18771	235868	576289
#Edge	4676	44327	1285465	30326273

Baselines. We compare HeFormer with three path-based methods: CN[18], AA[19] and RA[20], and seven GNN-based methods: GCN[21], SAGE[22], SEAL[9], Neo-GNN[23], NBFNet[24], ELPH[10] and BUDDY[10]. The results for SEAL, ELPH, and BUDDY are obtained by running the experiments on our local devices according to the original paper settings. Other results are taken from their respective studies or directly from the OGB leaderboard.

Evaluation Metrics. The rank of the positive link among the negative links is used to evaluate performance, calculating the proportion of positive test links ranked at or above the K-th position, which is HR@K. We use the metrics from the original studies, with HR@50 for ogbl-collab and HR@100 for the other datasets.

4.2 Main Results

We present the comparison results of HeFormer with baseline models on multiple benchmark datasets. The prediction accuracy results are shown in Table 2. We observe that HeFormer achieved state-of-the-art performance on three large datasets, with a notable 7% improvement on Pubmed. Additionally, HeFormer is the most stable among all methods, and no CUDA out-of-memory (OOM) errors occur.

Compared to the competing baseline BUDDY, HeFormer demonstrates a computational efficiency advantage, as shown in Table 3. Due to the small scale of Citeseer, the time differences are not significant. Our code is implemented using PyTorch Geometric [25] and PyTorch [26]. All experiments are conducted on servers equipped with four 24GB Quadro RTX 6000 GPUs. The runtime environment is kept consistent.

4.3 Ablation Study

We conducted ablation experiments to determine the effectiveness of generating structural features in HeFormer. We introduced two variants of HeFormer: (a) **w/o Node Feature**: retains structural features while removing node features; (b) **w/o Structure Feature**: retains node features while removing structural features. Fig. 5 presents the results of the ablation experiments. We observed that removing either feature always degrades performance, especially on ogbl-collab, where the accuracy dropped by 69% when structural features were not used.

Table 2. Results on link prediction benchmarks.

	Citeseer	Pubmed	Collab	PPA
CN	29.79 \pm 0.90	23.13 \pm 0.15	56.44 \pm 0.00	27.65 \pm 0.00
AA	35.19 \pm 1.33	27.38 \pm 0.11	64.35 \pm 0.00	32.45 \pm 0.00
RA	33.56 \pm 0.17	27.03 \pm 0.35	64.00 \pm 0.00	49.33 \pm 0.00
GCN	67.08 \pm 2.94	53.02 \pm 1.39	47.14 \pm 1.45	18.67 \pm 1.32
SAGE	57.01 \pm 3.74	39.66 \pm 0.72	54.63 \pm 1.12	16.55 \pm 2.40
Neo-GNN	84.67 \pm 2.16	73.93 \pm 1.19	62.13 \pm 0.58	49.13 \pm 0.60
NBFnet	74.07 \pm 1.75	58.73 \pm 1.99	OOM	OOM
SEAL	83.89 \pm 2.15	OOM	OOM	OOM
ELPH	89.66 \pm 0.82	66.83 \pm 0.58	OOM	OOM
BUDDY	89.25 \pm 0.14	71.22 \pm 0.78	65.73 \pm 0.58	45.74 \pm 0.88
HeFormer	88.57 \pm 0.41	76.20 \pm 0.46	65.89 \pm 0.21	50.74 \pm 0.64

Table 3. The feature dimension and the time (in seconds) taken to train each epoch.

Model	Dimensionality	Citeseer	Pubmed	Collab	PPA
ELPH	1024	0.2s	5.8s	OOM	OOM
BUDDY	1024	0.2s	0.6s	11.2s	202.0s
HeFormer	256	0.2s	0.6s	8.8s	181.0s

Max Hop Count of Node Neighborhoods. Fig. 5 also compares the results of HeFormer with different hop counts when constructing structural features. We found that while the accuracy of link prediction shows slight fluctuations with varying maximum hop counts, it remains higher than when structural features are removed.

Number of Lightweight Transformer Layers. We further analyze how changing the number of layers in the our Transformer affects link prediction accuracy and computational time. Fig. 6 presents the prediction accuracy and training time per epoch as the number of layers increases. Although previous studies have shown that multiple layers of Transformer usually aggregate more global information, leading to performance improvements, but this is not observed with HeFormer. We found that using more layers incurs a higher computational cost without significantly enhancing link prediction accuracy. This indicates that HeFormer, a single-layer, single-head Transformer, possesses excellent learning capability and computational efficiency.

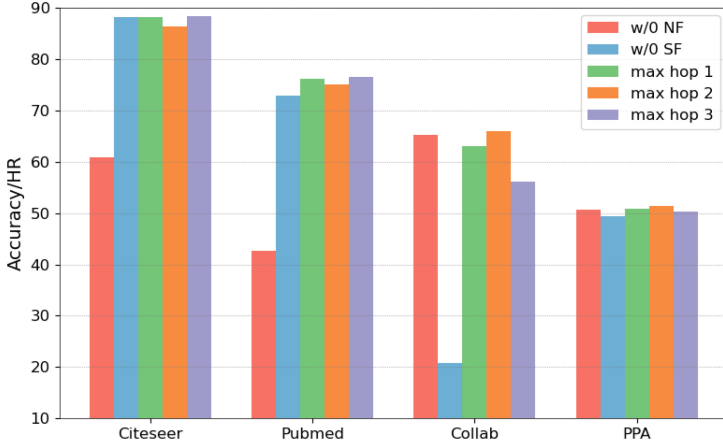


Fig. 5. Ablation experiment results showing the effects of removing node features (NF) or structure features (SF) from HeFormer. The results include the impact of varying the maximum hop number of the node neighborhoods when constructing the structural features.

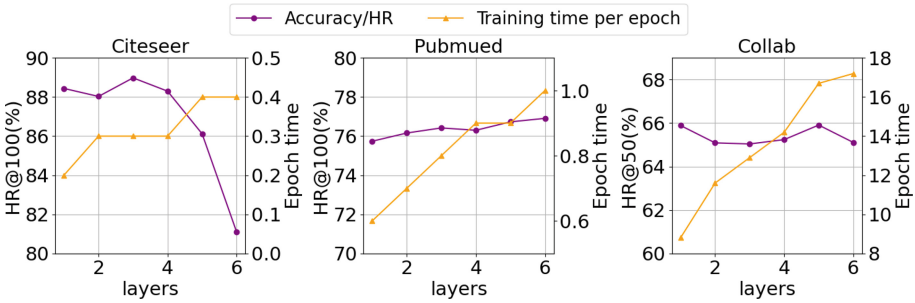


Fig. 6. Analyses of prediction accuracy and training time for HeFormer with different numbers of lightweight Transformer layers

5 Conclusions

In this work, we propose a link prediction model called HeFormer, which is based on hash-estimated structural features and a single-layer, single-head Transformer. Our model achieves efficient graph feature learning and message passing. Extensive experiments demonstrate that HeFormer can achieve state-of-the-art performance on large benchmark datasets while maintaining efficiency. We acknowledge that our parameter tuning is not optimal due to the diversity and scale of the datasets. Additionally, the advantage of global attention mechanisms in Transformers is less pronounced on smaller datasets. However, our work shows that shallow attention models can also achieve efficient learning by integrating subgraph structural information and node features. This points to a promising

new direction for building robust and lightweight Transformer for large-scale graph data processing tasks.

Acknowledgment. This work is supported by the National Science Foundation of China (No.61872372) and the Foundation of the State Key Laboratory of High Performance Computing (No.202101-11).

References

1. Zhang, X., Ning, W., Song, J., et al.: Tdlp: time decay based link prediction method for dynamic networks. In: International Conference on Electronic Information Engineering, Big Data, and Computer Technology (EIBDCT 2022). vol. 12256, pp. 633–639. SPIE (2022)
2. Zhang, X., Xiao, X., Li, G., Ning, W., Song, J.: Fig-lp: Feature-inverse-graph based link prediction in graph stream. In: 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta). pp. 1394–1401. IEEE (2022)
3. Feng, Z., Liu, L., Shu, J., Wang, P.: A survey of dynamic network link prediction. In: 2023 15th International Conference on Communication Software and Networks (ICCSN). pp. 143–147. IEEE (2023)
4. Louis, P., Jacob, S.A., Salehi-Abari, A.: Sampling enclosing subgraphs for link prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4269–4273 (2022)
5. Wu, Q., Zhao, W., Yang, C., Zhang, H., Nie, F., Jiang, H., Bian, Y., Yan, J.: Simplifying and empowering transformers for large-graph representations. In: Advances in Neural Information Processing Systems. vol. 36, pp. 64753–64773 (2023)
6. Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.Y.: Do transformers really perform badly for graph representation? In: Advances in Neural Information Processing Systems. vol. 34, pp. 28877–28888 (2021)
7. Wu, Z., Jain, P., Wright, M., Mirhoseini, A., Gonzalez, J.E., Stoica, I.: Representing long-range context for graph neural networks with global attention. In: Advances in Neural Information Processing Systems. vol. 34, pp. 13266–13279 (2021)
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30, pp. 5998–6008 (2017)
9. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Advances in Neural Information Processing Systems. vol. 31, pp. 5165–5175 (2018)
10. Chamberlain, B.P., Shirobokov, S., Rossi, E., Frasca, F., Markovich, T., Hammerla, N.Y., Bronstein, M.M., Hansmire, M.: Graph neural networks for link prediction with subgraph sketching. In: Proceedings of the 11th International Conference on Learning Representations (2023)
11. Pascoe, A.: Hyperloglog and minhash-a union for intersections. AdRoll, Apr **25**, 37 (2013)
12. Heule, S., Nunkesser, M., Hall, A.: Hyperloglog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm. In: Proceedings of the 16th International Conference on Extending Database Technology. pp. 683–692 (2013)
13. Wu, Q., Zhao, W., Li, Z., Wipf, D.P., Yan, J.: Nodeformer: A scalable graph structure learning transformer for node classification. In: Advances in Neural Information Processing Systems. vol. 35, pp. 27387–27401 (2022)

14. Srinivasan, B., Ribeiro, B.: On the equivalence between positional node embeddings and structural graph representations. In: International Conference on Learning Representations (2020)
15. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Advances in Neural Information Processing Systems. vol. 20, pp. 1177–1184 (2007)
16. Choromanski, K.M., Likhoshesterov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J.Q., Mohiuddin, A., Kaiser, L., Belanger, D.B., Colwell, L.J., Weller, A.: Rethinking attention with performers. In: International Conference on Learning Representations (2021)
17. 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. In: Advances in Neural Information Processing Systems. vol. 33, pp. 22118–22133 (2020)
18. Yao, L., Wang, L., Pan, L., Yao, K.: Link prediction based on common-neighbors for dynamic social network. *Procedia Computer Science* **83**, 82–89 (2016)
19. Adamic, L.A., Adar, E.: Friends and neighbors on the web. *Social networks* **25**(3), 211–230 (2003)
20. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. *The European Physical Journal B* **71**, 623–630 (2009)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
22. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. vol. 30, pp. 1024–1034 (2017)
23. Yun, S., Kim, S., Lee, J., Kang, J., Kim, H.J.: Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In: Advances in Neural Information Processing Systems. vol. 34, pp. 13683–13694 (2021)
24. Zhu, Z., Zhang, Z., Xhonneux, L.P., Tang, J.: Neural bellman-ford networks: A general graph neural network framework for link prediction. In: Advances in Neural Information Processing Systems. vol. 34, pp. 29476–29490 (2021)
25. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. In: International Conference on Learning Representations (2019)
26. Paszke, A., Gross, S., Massa, F., Lerer, et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. vol. 32, pp. 8024–8035 (2019)



Rethinking Attention Module Design for Point Cloud Analysis

Chengzhi Wu¹(✉), Kaige Wang¹, Zeyun Zhong¹, Hao Fu¹, Junwei Zheng¹,
Jiaming Zhang¹, Julius Pfrommer², and Jürgen Beyerer^{1,2}

¹ Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe,
Germany

{chengzhi.wu, zeyun.zhong, junwei.zheng, jiaming.zhang}@kit.edu,
{kaige.wang, hao.fu}@student.kit.edu

² Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB,
Karlsruhe, Germany

{julius.pfrommer, juergen.beyerer}@iosb.fraunhofer.de

Abstract. In recent years, there have been significant advancements in applying attention mechanisms to point cloud analysis. However, attention module variants featured in various research papers often operate under diverse settings and tasks, incorporating potential training strategies. This heterogeneity poses challenges in establishing a fair comparison among these attention module variants. In this paper, we address this issue by rethinking and exploring attention module design within a consistent base framework and settings. Both global-based and local-based attention methods are studied, with a focus on the selection basis and scales of neighbors for local-based attention. Different combinations of aggregated local features and computation methods for attention scores are evaluated, ranging from the initial addition/concatenation-based approach to the widely adopted dot product-based method and the recently proposed vector attention technique. Various position encoding methods are also investigated. Our extensive experimental analysis reveals that there is no universally optimal design across diverse point cloud tasks. Instead, drawing from best practices, we propose tailored attention modules for specific tasks, leading to superior performance on point cloud classification and segmentation benchmarks.

Keywords: Point cloud data · Attention mechanism · Module design exploration.

1 Introduction

The attention mechanism was first proposed by Bahdanau et al. [1] in 2014 to learn richer information from the input. Later, given its remarkable performance in the natural language processing domain [37] and 2D computer vision of image analysis [32], the research community also started to explore the application of attention modules for 3D point clouds. In 2021, Point Cloud Transformer (PCT) [5] and PT¹ [53] were the first to apply the attention mechanism to the point cloud learning tasks. In the following

years, several improvements have been made to the attention mechanism from different perspectives for better feature learning of point clouds [2, 7, 8, 10, 30, 33, 36, 50].

However, methods in different papers usually run under different settings and tasks, with potential various training tricks. Many papers claim that their proposed new modules achieve better performance, but the performance improvement may possibly be obtained due to the modifications in other parts. In this case, it is hard to determine which module is actually the optimal solution for one certain task. Hence, this work aims to conduct a comprehensive study on various attention module variants and provide a more equitable comparison, then propose more effective attention-based fundamental modules for different point cloud downstream tasks. Moreover, to better investigate how one attention module variant behaves under different downstream tasks, we adopt the same network model in different tasks for a fair comparison.

In this work, we explore the following four key aspects for attention module variants used in point cloud analysis: (1) neighbor selection operation; (2) local feature aggregation; (3) attention score computation methods; and (4) possible position encoding. Note that the former two aspects are only involved in local-based attention.

In neighbor selection operation for local-based learning, the measure of “distance” between points is mostly based on the coordinate distance or the feature difference between points. Apart from single-scale neighbor grouping, a multi-scale grouping strategy was also adopted in some papers. For example, Stratified Transformer [14] selects multi-scale neighbors through a stratified strategy for key sampling, while 3DCTN [21] implements multi-scale neighbor selection via a parallel multi-level multi-scale point transformer. For local feature aggregation, a combination of the following three types of features are often used: features of the centers, features of the neighbors, and the feature difference between the neighbor points and the center points [39].

Apart from the widely used dot product operation for computing attention scores, there are many other operation choices, including addition, concatenation, and subtraction. Each one also has more variants when considering global or local features. For example, PCT [5] adopts the commonly used Q K dot production to compute attention scores by only considering the global information, while PT¹ [53] uses Q K subtraction as the first step in local feature aggregation. The additive-based method was used in the original attention paper from Bahdanau et al. [1]. Another related research of Attention-based Neural Machine Translation [22] compared three attention approaches including multiplication, concatenation, and generalization.

Unlike text and images, point clouds are usually unordered, and thus the traditional position encoding should not be used to obtain better order variance of point inputs. In point cloud learning, position encoding simply means merging more information from the points’ 3D coordinates directly to the attention modules. For example, HiTPR [8] splices the difference between the coordinates of center and neighbor points with the difference of features, and then adds them to the attention map. LCPFormer [12] projects the original coordinates of the points to the same dimension as the structural information through MLP, and then adds them with the aggregated features. ProxyFormer [16] uses a self-attention operation to stitch the coordinates of local points with features and send them to the attention layer.

In this work, we summarize our core technical contributions concerning attention module design for point cloud analysis as follows:

- Selecting multi-scale neighbors as the Key input can mostly improve model performance yet model size and FLOPs increase significantly. To improve model performance with the same number of neighbors, grouping neighbors with a skipping strategy to achieve a larger perceptive field is recommended.
- In local feature aggregation, using the offset feature (the difference between the center point feature and the neighbor point feature) mostly yields a better result compared to using the neighbor feature directly.
- For global-based attention, L2-norm subtraction-based attention is overall better than the dot product self-attention. For local-based attention, offset-based attention modules achieve relatively better performance in both scalar and vector attention cases.
- Applying implicit position encoding is better than explicitly concatenating point coordinates to the attention input. Most implicit position encoding methods achieve similar favorable outcomes under various attention methods, and compressing its feature dimension during the encoding leads to less improved performance.
- We reveal that there is no such attention module that always achieves the best performance under different downstream tasks. However, some insights for choosing an optimal one are given through our exploration.

2 Related Work

Point Cloud Local Feature Aggregation. The pioneering work by Qi et al. introduces PointNet [26] for point cloud processing. Building upon it, PointNet++ [27] was introduced by incorporating local feature aggregation through a hierarchical neural network. Subsequently, DGCNN [39] introduces edge convolution to aggregate local features based on the k-nearest neighbors in the feature space. HiTPR [8] leverages local feature aggregation to enhance the model’s ability to recognize and match places in 3D environments. KPConv [35] introduces a deformable convolution operation for more flexible processing of irregularly distributed point cloud data. PointCNN [17] A novel X-convolution operation is proposed to better adapt to the disorder point clouds. PPT-Net [13] and StratifiedFormer [14] utilize a transformer-based architecture to capture fine-grained details and broader contextual information. MLMSPT [54] and 3DGTN [21] use a multi-scale neighbor point selection method to establish receptive fields of different scales and densities for extracting local features.

Attention-based Deep Learning on Point Clouds. By introducing the attention mechanism, models can selectively focus on important points or regions and effectively process point cloud data for various tasks [9, 19, 35, 40, 41]. PCT [5] is a typical method that applies the self-attention module directly to point cloud classification and segmentation tasks. On the other hand, cross-attention modules have been used to aggregate features from different regions of a point cloud, leading to more accurate and fine-grained predictions [3, 8, 25, 42, 43, 47, 48, 53]. PTTR [55] divides the input point cloud into multiple groups of points to extract their features respectively and then match and

predict these features. PT V2 [45] incorporates more efficient attention mechanisms and network architectures to boost performance while reducing computational demands.

Position Encoding in Point Cloud Analysis. The integration of position encoding in point cloud analysis has significantly enhanced model performance, as exemplified by Point Transformer [53]. Following this, PPTNet [13] further extends and refines the use of position encoding in point cloud processing. Liu et al. [51] propose a rotation-invariant position encoding method to ensure that the model performs stably on input point clouds in different orientations. RandLA-Net [9] explores the efficiency of position encoding in large-scale point cloud applications. Kan Wu et al. comparatively studied the effects of absolute position encoding [37] and relative position [11, 31, 34, 38] for attention mechanism. A contextual-based position encoding method is proposed in [44], and its variants have been widely used in point cloud learning tasks [14, 30, 45, 56].

3 Attention Module Variants

Methods in different papers run under different frameworks and settings, with potential various training tricks. To conduct a comprehensive study on various attention module variants and provide a more equitable comparison, we use an identical basic framework in all the experiments for one certain task, with the same setting and no special training tricks. The framework is given in Figure 1. It consists of an embedding layer, four sequential attention modules with residual links, and a task-oriented MLP head.

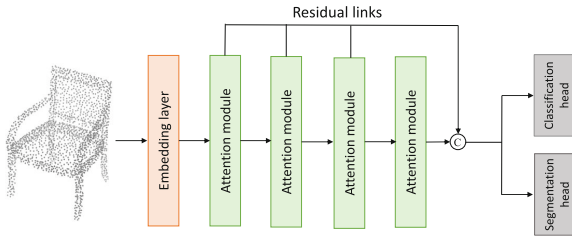


Fig. 1. Basic Framework. It consists of an embedding layer, four sequential attention modules with residual links, and a task-oriented MLP head.

The module differences of various attention module variants can be summarized in the following four aspects: neighbor selection, local feature aggregation, position encoding, and Q K V feature fusion method (i.e., the actual attention method). Note that the former two aspects are only involved in local-based attention.

3.1 Neighbor Selection

When considering local information, the first decision to make is on what basis the neighbors are selected, i.e., on original point 3D coordinates, or the feature similarity in the high-dimensional feature space. After the basis is determined, k neighbors are

selected with the K-Nearest Neighbors (KNN) method in the vanilla case. However, it is possible to select the same number of neighbor points with a larger perceptual field by regular skipping. As illustrated in Figure 2(a), for scale α , $k * 2^\alpha$ nearest neighbors are first obtained, then k points are selected with a step of 2^α .

Moreover, it is possible to consider the case of multi-scale, i.e., use the neighbor groups of different perceptual field sizes as multiple keys for the attention operation. For multi-scale as separate keys, the number of keys in the attention layer is equal to the number of scales. Each scale selects points from the same starting point using different degrees of sparsity. For multi-scale as one key, there is no overlapping between different scales. As illustrated in Figure 2(b), to avoid repetitive point selection, multi-scale as separate keys method obtains $k * (2^{\alpha+1} - 1)$ nearest neighbors first, then select k points with different steps in different segments.

3.2 Local Feature Aggregation

In local-based attention, the features that can be used for local feature aggregation include (i) the feature of the center point; (ii) the feature of selected neighbor points; and (iii) the offset feature between neighbor points and the center point. Different combinations of these three methods are tested for different attention methods. K should include at least one of the neighbor feature and the offset feature.

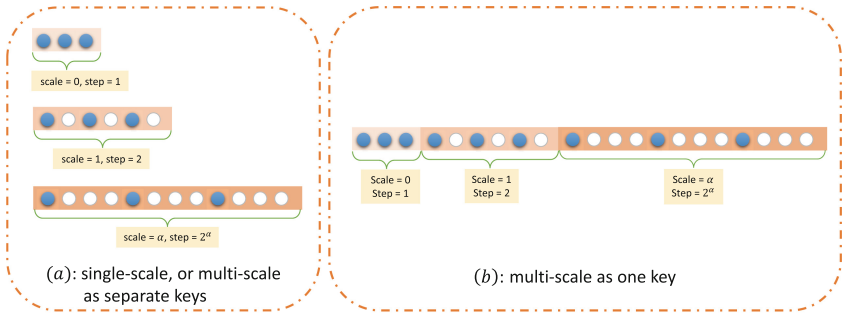


Fig. 2. (a) Single-scale, or multi-scale as separate keys, and (b) multi-scale as one key. A sparser point selection method is used in larger perceptual fields, with the number of points selected in each scale being consistent.

It is worth noting that the attention score computation method influences the choice of features used for local feature aggregation. For example, when using offset/subtraction-based attention methods to compute attention scores, the operation already contains the offset feature information implicitly (Q contains the center feature, while K contains the neighbor feature). Hence the local feature aggregation part should exclude the feature differences in this case. Another example is that when using addition-based attention methods to compute attention scores, the local feature aggregation part should exclude the center feature. A corresponding table is given in Table 1. Detailed introductions of various attention operation methods are given in Section 3.3.

Table 1. Possible combinations of features for local feature aggregation in different attention operation methods.

Att. method	Local fea.		center, center, neighbor,		center, neighbor, offset	
	neighbor	offset	neighbor	offset	neighbor,	offset
Dot product	✓	✓	✓	✓	✓	✓
Offset/Subtraction	✓	-	✓	-	-	-
Addition	✓	✓	-	-	✓	-
Concat	✓	✓	-	-	✓	-

3.3 Attention Method

Based on whether local information is considered, the attention methods used for point cloud analysis can be mainly divided into two categories: global-based attention, and local-based attention.

Global-based Attention Global-based attention module is designed as shown in Figure 3. It can be described as \mathcal{F}_g :

$$\mathcal{F}_g = \text{FFN}(\phi^g(p) + p) \tag{1}$$

where ϕ^g denotes the global self-attention function, and p represents the point cloud features and is the input of the entire module. A residual link is used to convey the input to the post-attention tensor. Finally, the output is obtained through a feedforward neural network (FFN).

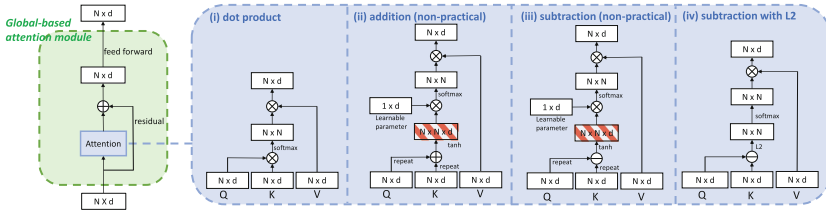


Fig. 3. Global-based attention module.

Dot Product. For global-based attention modules, ϕ^g in Equation (1) is regarded as the main research object. Q K dot product as shown in Figure 3(i) is widely used as a calculation method for attention scores [7, 10, 30, 33, 36, 50], and it can be described by the following function:

$$\phi_{dot}^g = \text{Softmax} \left(\frac{Q \cdot K}{\sqrt{d}} \right) \cdot V \tag{2}$$

Direct Addition and Subtraction. The computational methods of addition [1] and subtraction can also be used to compute attention scores. However as shown in Figure 3

(ii) and (iii), the problem of too large size tensor ($N \times N \times d$) arises in the computational procedure, which makes the methods actually inapplicable.

Subtraction with L2. On the other hand, for global attention, L2 norm-based subtraction is applicable by employing mathematical equivalence calculation tricks. PS-Former [2] use of subtraction with L2 distance to realize the calculation of attention scores, reduces the transmission dimension of the subtraction operation. The subtraction combined with L2 shown in Figure 3(iv) is described as:

$$\phi_{L2}^g = \text{softmax}\left(\frac{-\|Q - K\|_2^2}{\sqrt{d}}\right) \cdot V \quad (3)$$

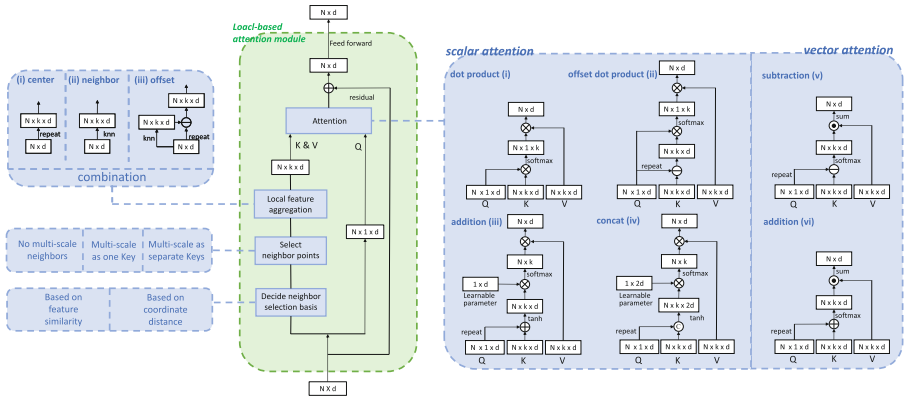


Fig. 4. Local-based attention module.

Local-based Attention Local-based attention module is designed as follows shown in Figure 4, It can be described as \mathcal{F}_l :

$$\mathcal{F}_l = \text{FFN}(\phi^l(p, \mathcal{G}(p, \mathcal{N}(p)) + p) \quad (4)$$

where \mathcal{N} denotes the neighbor selection method, \mathcal{G} denotes the local feature aggregation method, and ϕ^l represents the local feature-based cross-attention.

For local-based attention module, neighbor selection function \mathcal{N} in Equation (4) will be based on the difference of features between points \mathcal{N}^f or the difference of coordinates \mathcal{N}^c . Neighbor points are selected based on KNN. For each point, When it is used as the center point for neighbor selection, neighbor points of multi-scale could possibly be selected. The multi-scale can be divided into two structures, multi-scale as one key \mathcal{N}_{one} shown in Figure 2(a) and multi-scale as separate keys \mathcal{N}_{sep} shown in Figure 2(b).

For local feature aggregation functions \mathcal{G} , as shown in Figure 4 local feature aggregation block. the features that can be used for local feature aggregation include (i) the

feature of center points; (ii) the feature of k neighbor points; and (iii) the feature difference between neighbor points and center points. we will test different combinations of these three methods. It is worth noting that there is an influence between the choice of aggregation methods and the attention score computation method.

Dot Product. For cross attention function ϕ^l , similar to global attention, dot product shown in Figure 4(i) is widely used in local-based cross attention score computation:

$$\phi_{dot}^l = \text{softmax} \left(\frac{Q \cdot K}{\sqrt{d}} \right) \cdot V \quad (5)$$

Offset dot Product. For the dot product method, if the input is the difference between neighbor points and center points(offset feature), the attention module performs the “offset before MLP” operation. In the meta-base model [18], Lin et al. completed a comparative experiment on whether to perform MLP before Group [14, 17, 20, 53] or Group before MLP [15, 26, 29, 39]. Inspired by this experiment, as a comparison with dot product, dot product with offset shown in Figure 4(ii) is based on the dot product setting of MLP before offset:

$$\phi_{dot''}^l = \text{softmax} \left(\frac{Q \cdot (Q_r - K)}{\sqrt{d}} \right) \cdot V \quad (6)$$

Addition. When attention mechanism was proposed, the computational method of addition was used. we follow this method as one of the attention variants. [1]. addition method as shown in Figure 4(iii) [1] is given:

$$\phi_{add}^l = \text{softmax}(\omega^\top \cdot \tanh(Q_r + K)) \cdot V \quad (7)$$

where Q_r means that Q repeats K times in the dimension of the number of neighbor points, ω^\top is a learnable parameter, which enhances the expressive ability of the model to a certain extent while making parameter transmission smoother and reduces information loss.

Concatenation. Similarly, using the concatenation operation on Q K and implementing the concatenation method as shown in Figure 4(iv) with the help of the learnable parameter w , it can be expressed as:

$$\phi_{cat}^l = \text{softmax}(\omega^\top \cdot \tanh(\text{concat}(Q_r, K))) \cdot V \quad (8)$$

Vector attention with subtraction. Inspired by the use of vector attention mechanisms in image processing [52], PT2[45] uses advanced vector attention mechanisms in point clouds. Considering the rationality of feature dimension transformation, we designed vector attention functions based on subtraction shown in Figure 4(v) and addition shown in Figure 4(vi), which can be described as:

$$\phi_{v-sub}^l = \text{softmax}(Q_r - K) \odot V \quad (9)$$

where Q_r means that Q repeats k times in the dimension of the number of neighbor points, \odot is the Hadamard product.

Vector attention with addition. Since vector attention is less explored in point cloud deep learning, we additionally test the following addition-based vector attention method:

$$\phi_{v-add}^l = \text{softmax}(Q_r + K) \odot V \quad (10)$$

In the following experiments, the above attention score computation methods are tested with appropriate aggregated features, as listed in Table 4.

3.4 Position Encoding

We explore four widely-used position encoding methods, as illustrated in Figure 5. The first one explicitly concatenates the point coordinates with the learned latent representation from the last layer. The other three methods learn separate MLP projections and add the information to certain joints of attention modules.

method (i). Based on the original self-attention which considers the absolute position [37], method (i) δ_1 directly concatenates the spatial coordinates to the Q, K, V input. This ensures the spatial information is explicitly merged with the attention input, thereby enhancing the network’s ability to leverage absolute position information.

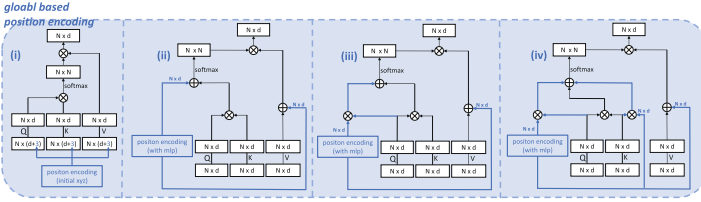


Fig. 5. Position encoding in global-based attention.

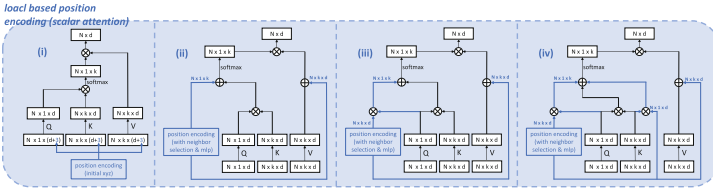


Fig. 6. Position encoding in local-based scalar attention.

method (ii). Method (ii) introduces an implicit encoding with an MLP that encodes positional information. This method relies on the network’s ability to infer spatial relationships implicitly:

$$\delta_2 = (Q \cdot K + \text{MLP}(p_{xyz})) \cdot (V + \text{MLP}(p_{xyz})) \quad (11)$$

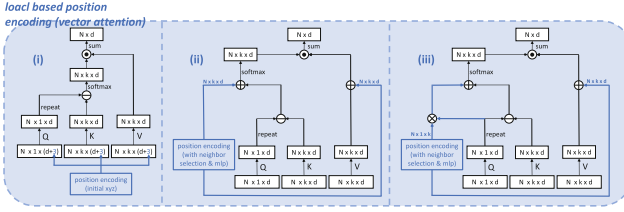


Fig. 7. Position encoding in local-based vector attention.

method (iii). Method (iii) enhances the implicit positional encoding by integrating it with the Q matrix. The encoding is designed to include contextual information derived from the data points’ relationships in the Q space. This method uses an MLP to process the initial positional information:

$$\delta_3 = (Q \cdot K + Q \cdot \text{MLP}(p_{xyz})) \cdot (V + \text{MLP}(p_{xyz})) \quad (12)$$

method (iv). Method (iv) goes a step further by incorporating contextually enriched positional information into both the Q, K matrices. The MLP here processes the positional information and integrates it with Q and K, facilitating a more detailed comparison between points during the attention calculation:

$$\delta_4 = (Q \cdot K + Q \cdot \text{MLP}(p_{xyz}) + K \cdot \text{MLP}(p_{xyz})) \cdot (V + \text{MLP}(p_{xyz})) \quad (13)$$

The position encoding operation under the local attention mechanism is shown in Figure 6 and Figure 7. There are differences in its dimensional transformation and global attention contrast, which will affect the performance of position encoding. This will be discussed in detail in Section 4.4.

4 Explore Best Practices for Different Tasks

4.1 Experiment setting

In this paper, two widely-used tasks are used as benchmarks for the experiments of attention module variants: point cloud classification on ModelNet40 [46], and point cloud segmentation on ShapeNetPart [49].

Datasets ModelNet40 dataset contains 12,311 pre-aligned shapes from 40 categories, which are split into 9,843 (80%) for training and 2,468 (20%) for testing. ShapeNetPart dataset contains 16,881 pre-aligned shapes from 16 categories, annotated with 50 segmentation parts in total. Most object categories are labeled with two to five segmentation parts. There are 12,137 (70%) shapes for training, 1,870 (10%) shapes for validation, and 2,874 (20%) shapes for testing.

Training Details For both classification and segmentation experiments, our setup involved training 200 epochs with a batch size of 16, splitting over 2x GTX 2080Ti for standard tasks, or 2x RTX 3090 for some multi-scale configuration experiments which need a larger memory. AdamW is used as the optimizer. The learning rate starts

from 1×10^{-4} and decays to 1×10^{-8} with a cosine annealing schedule. A warp-up strategy is used for more stable performance. The weight decay for model parameters is set as 1 for classification experiments and 1×10^{-4} for segmentation experiments. We augmented the inputs by randomly applying the following four methods: jittering, rotation, translation, and anisotropic scaling.

4.2 Neighbor Selection

As shown in Table 3, different multi-scale selection strategies are applied when selecting neighbor points. In this part of the experiment, we set the attention score calculation method to dot product, and the aggregation feature is the offset feature. First, an appropriate number of neighbor points k needs to be determined under different tasks. We tested the performance of the model with different numbers of neighbor points as shown in Table 2. In the classification task, $k = 32$ obtained the best performance. In the segmentation task, a larger k achieves better performance. Taking the computational complexity and performance into account, we use $k = 32$ for most of the following experiments.

Table 2. Classification and segmentation performance of using different k with all other settings consistent. The applied attention method is local dot product. The offset feature is used for local feature aggregation. Only single scale $\alpha = 0$ is used.

	k	4	8	16	32	64	128
Cls. OA (%)		92.49±0.11	92.74±0.09	92.89±0.11	93.10±0.06	92.97±0.08	92.84±0.06
Seg. Cat. mIoU (%)		84.04±0.08	84.90±0.10	85.53±0.08	85.59±0.06	85.90±0.07	86.29±0.06
	Ins. mIoU (%)	79.58±0.10	80.84±0.06	82.03±0.07	82.60±0.06	83.10±0.05	84.05±0.07

As shown in Table 3, multi-scale as one key with scales $\alpha = \{0, 1, 2\}$ achieves the best results in downstream tasks. a larger-scale point selection method can provide richer contextual information, which can help the model better generalize to different scenarios and conditions and improve model performance. On the other hand, the strategy of using separate keys does not exhibit any discernible impact, yet it requires much more model parameters. Hence we can conclude that using separate attention modules for separate scales (e.g. 3DCTN [21]) is not necessary, using multi-scale as one key is an overall better solution (e.g. Stratified Transformer [14]). Moreover, if certain limitations of computational resource pose, single scale yet with a larger perceptive field (larger scale α) for neighbor searching is also a practical solution to reduce the FLOPs of the model while still achieving decent performance.

4.3 Local Feature Aggregation and Attention Method

As mentioned in Section 3.2, in the local-based attention module, the attention score calculation method affects the selection of features for local feature aggregation. Both

Table 3. Classification and segmentation performance of different variants under different neighbor selection basis and scales. The parameters and FLOPs of the attention module are also reported.

Variants			Cls. OA (%)	Seg.		Params (k)	FLOPs (G)
Neighbor Basis	Scale	One/sep. Keys		Cat. mIoU (%)	Ins. mIoU (%)		
Feature Similarity	0	one	93.10±0.09	82.60±0.06	85.59±0.07	180.22	2.45
	1	one	93.22±0.07	82.87±0.08	85.74±0.04	180.22	2.45
	2	one	93.30±0.07	83.02±0.07	85.92±0.11	180.22	2.45
	0, 1	one	93.27±0.06	82.97±0.05	85.95±0.02	180.22	4.60
	0, 1	sep	93.25±0.05	82.90±0.05	85.85±0.06	229.38	4.63
	0, 1, 2	one	93.34±0.07	82.96±0.09	85.97±0.09	180.22	6.75
	0, 1, 2	sep	93.34±0.04	83.15±0.06	85.76±0.10	278.53	6.81
	Initial 3D Coordinates	0	one	92.32±0.09	82.26±0.10	84.22±0.03	180.22
1		one	92.54±0.06	82.41±0.09	85.05±0.05	180.22	2.45
2		one	92.70±0.08	82.51±0.05	85.29±0.07	180.22	2.45
0, 1		one	92.75±0.03	82.55±0.06	85.21±0.06	180.22	4.60
0, 1		sep	92.74±0.06	82.54±0.06	85.13±0.07	229.38	4.63
0, 1, 2		one	92.93±0.07	82.78±0.07	85.25±0.06	180.22	6.75
0, 1, 2		sep	92.90±0.05	82.96±0.03	85.19±0.04	278.53	6.81

Table 4. Classification and segmentation performance of different variants under different attention score computation methods and corresponding used features for local feature aggregation.

Variants			Cls. OA (%)	Seg.		Params (k)	FLOPs (G)
Global/ Local	Attention	Agg.		Cat. mIoU (%)	Ins. mIoU (%)		
Global	Dot Product	-	93.02±0.05	83.07±0.08	85.45±0.07	180.22	0.37015
	Subtraction	-	93.43±0.07	83.15±0.06	85.47±0.06	180.22	0.37015
Local	Scalar Dot Product	Neighbor	92.98±0.08	82.24±0.08	85.48±0.04	180.22	2.45
		Offset	93.10±0.09	82.60±0.06	85.59±0.07	180.22	2.45
		Center, Neighbor	92.98±0.05	83.03±0.08	85.25±0.05	212.99	4.60
		Center, Offset	93.26±0.07	83.04±0.05	85.60±0.07	212.99	4.60
		Neighbor, Offset	93.14±0.06	83.15±0.04	85.57±0.06	212.99	4.60
		Center, Neighbor, Offset	93.20±0.08	83.23±0.04	85.74±0.06	245.76	6.75
		Center, Neighbor	92.78±0.04	82.85±0.06	85.60±0.08	212.99	4.60
	Scalar Offset Dot Product	Neighbor	93.30±0.09	82.77±0.06	85.37±0.09	180.22	2.45
		Center, Neighbor	92.78±0.04	82.85±0.06	85.60±0.08	212.99	4.60
		Neighbor	92.65±0.07	82.15±0.11	85.37±0.06	180.22	2.45
	Scalar Addition	Offset	93.38±0.05	82.71±0.08	85.54±0.04	180.22	2.45
		Neighbor, Offset	93.30±0.07	82.94±0.09	85.54±0.07	212.99	4.60
		Neighbor	93.38±0.07	82.86±0.05	85.40±0.05	180.22	2.45
	Scalar Concat	Offset	93.51±0.09	83.20±0.04	85.53±0.05	180.22	2.45
		Neighbor, Offset	93.06±0.06	82.94±0.04	85.57±0.09	212.99	4.60
		Neighbor	93.55±0.04	82.57±0.03	85.44±0.07	180.22	2.45
	Vector Subtraction	Center, Neighbor	92.94±0.04	83.24±0.05	85.50±0.07	212.99	4.60
		Neighbor	93.14±0.09	82.89±0.03	85.74±0.07	180.22	2.45
	Vector Addition	Offset	93.06±0.07	82.84±0.03	85.43±0.11	180.22	2.45
		Neighbor, Offset	93.06±0.10	82.95±0.05	85.76±0.06	212.99	4.60

aspects must be assessed simultaneously. For a more comprehensive comparison, this subsection also includes experimental results of the global-based attention module. As shown in Table 1, different attention score computation methods correspond to a variety of reasonable feature aggregation methods. Under the configuration of scale $\alpha = 0$ and $k = 32$, the experimental results are shown in Table 4.

From the table, we can observe that for attention methods with which neighbor feature and offset feature are both applicable, using offset feature achieves better performance than using neighbor feature. Moreover, for classification tasks, when offset feature is already used, adding neighbor feature additionally results in a modest decrease in performance. This shows that the following two principles can enable the model to perform well in classification tasks: (i) the input information contains offset features, (ii) the redundancy of the aggregated information is reduced. However, in segmentation tasks, feature aggregation methods with higher redundancy can mostly achieve better performance. But this also brings higher FLOPs as shown in Table 4. This is a trade-off that needs to be balanced in the actual application scenarios.

In the classification task, the local-based vector offset attention ϕ_{v-sub}^l with neighbor feature aggregated achieves the best performance. In the segmentation task, the local-based vector offset attention ϕ_{v-sub}^l with center and neighbor features aggregated achieves the best performance. This provides insights for possible improvements to the existing models, e.g., both center and neighbor features can be used in HitPR [8] for getting better task performance.

On the other hand, we find that the global subtraction attention method also achieves decent performance with much smaller FLOPs. And the L2-norm subtraction-based attention ϕ_{L2}^g is overall better than the dot product self-attention. It should be pointed out that although the FLOPs difference between the local and global attention modules is very large, with the embedding layer and task-oriented MLP head, and the actual training time difference between the two kinds of attention modes is not as large as that of FLOPs. Under the configuration of Section 4.1, local-based and global-based methods take around 13 hours and 7 hours to complete the training respectively for the classification task, and 19 hours and 10 hours for the segmentation task.

4.4 Position Encoding

The experimental results are reported in Table 5 and it indicates that different position encoding methods impact model performance in classifying and segmenting point cloud data. Explicit position encoding (δ_1), which adds spatial coordinates directly, does improve classification accuracy to some extent, but its effectiveness is limited in complex segmentation tasks, suggesting that relying solely on absolute spatial information is insufficient for handling intricate point relationships. In contrast, implicit position encoding (δ_2 , δ_3 and δ_4) shows greater advantages, especially when combined with contextual information. By integrating rich contextual information with the Query and Key input, δ_3 and δ_4 significantly enhance the model’s performance in both classification and segmentation tasks, emphasizing the importance of a global perspective in understanding point relationships. These findings highlight that implicit position encoding with contextual information strategy is more effective for downstream tasks.

Table 5. Classification and segmentation performance obtained by selecting different position encoding methods under different attention methods. PE stands for position encoding.

Variants				Cls. OA (%)	Seg.		Params (k)	FLOPs (G)
Global/ Local	Attention	Agg.	PE		Cat. mIoU (%)	Ins. mIoU (%)		
Global	Dot Product	-	-	93.02±0.05	83.07±0.08	85.45±0.07	180.22	0.37015
			i	93.18±0.10	82.90±0.08	85.29±0.09	181.38	0.37251
			ii	93.06±0.07	83.11±0.07	85.35±0.07	180.62	0.37096
			iii	93.18±0.07	83.22±0.04	85.48±0.06	180.99	0.37172
	iv	93.22±0.03	83.18±0.07	85.53±0.07	181.38	0.37251		
	Subtraction	-	-	93.43±0.07	83.15±0.06	85.47±0.06	180.22	370.15
			i	93.38±0.08	83.21±0.08	85.55±0.07	181.38	0.37251
			ii	93.34±0.09	83.12±0.09	85.48±0.07	180.62	0.37096
iii			93.51±0.04	83.16±0.08	85.60±0.06	180.99	0.37172	
iv	93.46±0.05	83.19±0.05	85.64±0.06	181.38	0.37251			
Local	Scalar Dot Product	Offset	-	93.10±0.09	82.60±0.06	85.59±0.07	180.22	2.45
			i	93.14±0.08	82.73±0.09	85.52±0.07	181.38	2.50
			ii	93.14±0.06	82.85±0.07	85.58±0.06	180.75	2.48
			iii	93.30±0.06	82.92±0.07	85.59±0.03	181.25	2.50
	iv	93.22±0.06	83.05±0.04	85.77±0.03	181.76	2.50		
	Scalar Offset Dot Product	Neighbor	-	93.30±0.09	82.77±0.06	85.37±0.09	180.22	2.45
			i	92.25±0.07	82.85±0.09	85.43±0.10	181.38	2.50
			ii	93.22±0.08	82.84±0.07	85.54±0.08	180.75	2.48
			iii	93.24±0.05	83.15±0.05	85.60±0.05	181.25	2.50
	iv	93.55±0.04	82.92±0.08	85.72±0.03	181.76	2.50		
	Vector Subtraction	Neighbor	-	93.55±0.04	82.57±0.03	85.44±0.07	180.22	2.45
			i	93.47±0.05	82.62±0.05	85.42±0.06	181.38	2.50
ii			93.56±0.05	83.26±0.04	85.55±0.05	181.25	2.50	
iii			93.32±0.06	82.54±0.09	85.29±0.07	180.75	2.48	

It is worth noting that when local-based attention is used, the power of position encoding is possibly limited by the feature dimension decrease in the MLPs that are used for encoding. For example, in δ_2 of Figure 6 and δ_3 of Figure 7, the feature dimension of point positions has to be mapped from 3 to 1 to satisfy the architecture, resulting in less improved performance. This is also the reason why it is difficult for δ_2 in the local scalar attention and δ_3 in the local vector attention to improve model performance.

5 Apply Best Practices

Through the data analysis in Section 4, The best modules under global and local attention mechanisms will be selected for testing. For module selection, we are prioritizing the best performance while also taking into account computational efficiency. Considering that downstream tasks of different complexity, we use different frameworks to handle classification and segmentation tasks.

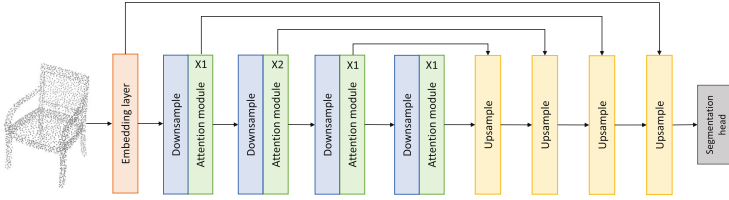


Fig. 8. The segmentation network architecture for applying best practices.

5.1 Classification

For the classification tasks, we use the basic framework shown in Figure 1 to test on ScanObjectN and ModelNet40 benchmarks. By applying the best practices explored before, the following two attention module variants are used: for global-based attention, we use L2-norm subtraction-based attention method ϕ_{L2}^g with position encoding method δ_3 ; for local-based attention, we use subtraction-based vector attention ϕ_{v-sub}^l with neighbor feature aggregated and with position encoding method δ_2 . The results are shown in Table 6. Our experimental results are better than or on par with the state-of-the-art methods. Moreover, please note that we achieve such superior performance with a relatively smaller number of parameters and FLOPs.

Table 6. 3D object classification performance on ScanObjectNN and ModelNet40. The parameters and FLOPs of the entire framework are also reported.

Method	ScanObjectNN(PB_T50_RS)		ModelNet40		Params (M)	FLOPs (G)
	OA(%)	mAcc(%)	OA(%)	mAcc(%)		
PointNet [26]	68.2	63.4	89.2	86.2	3.5	0.9
PointCNN [17]	78.5	75.1	92.2	88.1	0.6	-
DGCNN [39]	78.1	73.6	92.9	90.2	1.8	4.8
DeepGCN [15]	-	-	93.6	90.9	2.2	3.9
KPConv [35]	-	-	92.9	-	14.3	-
ASSANet-L [28]	-	-	92.9	-	118.4	-
SimpleView [4]	80.5±0.3	-	93.0±0.4	90.5±0.8	0.8	-
MVTN [6]	82.8	-	93.5	91.2	3.5	1.8
PCT [5]	-	-	93.2	-	2.9	2.3
CurveNet [24]	-	-	93.8	-	2.0	-
PointMLP [23]	85.4	83.9	94.1	91.3	13.2	31.3
Ours (global)	83.1±0.4	80.8±0.6	93.8±0.1	90.7±0.2	1.93	3.67
Ours (local)	83.7±0.5	81.2±0.9	93.9±0.2	91.1±0.3	1.93	7.70

5.2 Segmentation

A framework illustrated in Figure 8 is used for the segmentation task. By applying the best practices explored before, the following two attention module variants are used: for global-based attention, we use L2-norm subtraction-based attention method ϕ_{L2}^g with position encoding method δ_4 ; for local-based attention, we use offset dot product-based scalar attention ϕ_{dot}^l with neighbor feature aggregated and with position encoding method δ_4 . Data is progressively downsampled and processed, followed by upsampling modules to increase data resolution. The results are reported in Table 7. Note that the FLOPs number is drastically decreased since the segmentation framework has multiple downsample layers. Despite not reaching the level of state-of-the-art methods, our framework still demonstrates relatively superior performance when considering parameter scaling and computational complexity. It achieves such performance with a much smaller number of FLOPs, highlighting the efficiency and effectiveness of our attention module choices.

Table 7. Part segmentation performance on ShapeNet Part.

Method	Cat. mIoU (%)	Ins. mIoU (%)	Params. (M)	FLOPs (G)
PointNet [26]	80.4	83.7	3.6	4.9
DGCNN [39]	82.3	85.2	1.3	12.4
KPCConv [35]	85.1	86.4	-	-
CurveNet [24]	-	86.8	-	-
ASSANet-L [28]	-	86.1	-	-
Point Cloud Transformer [5]	83.7	86.6	7.8	-
PointMLP [23]	84.6	86.1	-	-
Stratifiedformer [14]	85.1	86.6	-	-
Ours (global)	84.25±0.10	86.27±0.09	5.0	0.64
Ours (local)	84.37±0.07	86.36±0.05	5.0	1.39

6 Conclusion

In this paper, we conduct an extensive and fair comparative study of attention mechanisms under a unified framework and summarize some best practices in the attention module design for point cloud analysis. Furthermore, we follow these best practices and propose to use different attention modules for different downstream tasks and achieve good performance and efficiency. In summary, rethinking the attention mechanism helps to clarify the characteristic differences between different attention module variants for point cloud analysis, and provides important insights for the design and exploration of future network architectures.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
2. Ding, Z., Hou, J., Tu, Z.: Point cloud recognition with position-to-structure attention transformers. arXiv preprint [arXiv:2210.02030](https://arxiv.org/abs/2210.02030) (2022)
3. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer. *IEEE Access* **9**, 134826–134840 (2021)
4. Goyal, A., Law, H., Liu, B., Newell, A., Deng, J.: Revisiting point cloud shape classification with a simple and effective baseline. In: *ICML*. pp. 3809–3820 (2021)
5. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: PCT: Point cloud transformer. *Computational Visual Media* **7**, 187–199 (2021)
6. Hamdi, A., Giancola, S., Ghanem, B.: MVTN: Multi-view transformation network for 3d shape recognition. In: *ICCV*. pp. 1–11 (2021)
7. Han, X.F., Jin, Y.F., Cheng, H.X., Xiao, G.Q.: Dual transformer for point cloud analysis. *IEEE Transactions on Multimedia* (2022)
8. Hou, Z., Yan, Y., Xu, C., Kong, H.: HiTPR: Hierarchical transformer for place recognition in point cloud. In: *ICRA*. pp. 2612–2618. *IEEE* (2022)
9. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Rاندلانet: Efficient semantic segmentation of large-scale point clouds. In: *CVPR*. pp. 11108–11117 (2020)
10. Huang, Q., Dong, X., Chen, D., Zhou, H., Zhang, W., Zhang, K., Hua, G., Cheng, Y., Yu, N.: PointCAT: Contrastive adversarial training for robust point cloud recognition. *IEEE Transactions on Image Processing* (2024)
11. Huang, Z., Liang, D., Xu, P., Xiang, B.: Improve transformer models with better relative position embeddings. arXiv preprint [arXiv:2009.13658](https://arxiv.org/abs/2009.13658) (2020)
12. Huang, Z., Zhao, Z., Li, B., Han, J.: LCPFormer: Towards effective 3d point cloud analysis via local context propagation in transformers. *IEEE Transactions on Circuits and Systems for Video Technology* (2023)
13. Hui, L., Yang, H., Cheng, M., Xie, J., Yang, J.: Pyramid point cloud transformer for large-scale place recognition. In: *ICCV*. pp. 6098–6107 (2021)
14. Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J.: Stratified transformer for 3d point cloud segmentation. In: *CVPR*. pp. 8500–8509 (2022)
15. Li, G., Muller, M., Thabet, A., Ghanem, B.: DeepGCNs: Can GCNs go as deep as CNNs? In: *ICCV*. pp. 9267–9276 (2019)
16. Li, S., Gao, P., Tan, X., Wei, M.: ProxyFormer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In: *CVPR*. pp. 9466–9475 (2023)
17. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: PointCNN: Convolution on x-transformed points. *NeurIPS* **31** (2018)
18. Lin, H., Zheng, X., Li, L., Chao, F., Wang, S., Wang, Y., Tian, Y., Ji, R.: Meta architecture for point cloud analysis. In: *CVPR*. pp. 17682–17691 (2023)
19. Liu, X., Han, Z., Liu, Y.S., Zwicker, M.: Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In: *AAAI*. vol. 33, pp. 8778–8785 (2019)
20. Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X.: A closer look at local aggregation operators in point cloud analysis. In: *ECCV*. pp. 326–342 (2020)
21. Lu, D., Xie, Q., Gao, K., Xu, L., Li, J.: 3DCTN: 3d convolution-transformer network for point cloud classification. *IEEE Trans. Intell. Transp. Syst.* **23**(12), 24854–24865 (2022)
22. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint [arXiv:1508.04025](https://arxiv.org/abs/1508.04025) (2015)

23. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. arXiv preprint [arXiv:2202.07123](https://arxiv.org/abs/2202.07123) (2022)
24. Muzahid, A., Wan, W., Sohel, F., Wu, L., Hou, L.: CurveNet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica* **8**(6), 1177–1187 (2020)
25. Park, J., Lee, S., Kim, S., Xiong, Y., Kim, H.J.: Self-positioning point-based transformer for point cloud understanding. In: *CVPR*. pp. 21814–21823 (2023)
26. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In: *CVPR*. pp. 652–660 (2017)
27. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS* **30** (2017)
28. Qian, G., Hammoud, H., Li, G., Thabet, A., Ghanem, B.: ASSANet: An anisotropic separable set abstraction for efficient point cloud representation learning. *NeurIPS* **34**, 28119–28130 (2021)
29. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS* **35**, 23192–23204 (2022)
30. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. In: *CVPR*. pp. 11143–11152 (2022)
31. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. *NeurIPS* **32** (2019)
32. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: *ICCV*. pp. 12179–12188 (2021)
33. Shan, J., Zhou, S., Fang, Z., Cui, Y.: PTT: Point-track-transformer module for 3d single object tracking in point clouds. In: *IROS*. pp. 1310–1316 (2021)
34. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. arXiv preprint [arXiv:1803.02155](https://arxiv.org/abs/1803.02155) (2018)
35. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: KPConv: Flexible and deformable convolution for point clouds. In: *ICCV*. pp. 6411–6420 (2019)
36. Umam, A., Yang, C.K., Chuang, Y.Y., Chuang, J.H., Lin, Y.Y.: Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In: *ECCV*. pp. 596–611 (2022)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *NeurIPS* **30** (2017)
38. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: *ECCV*. pp. 108–126 (2020)
39. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics* **38**(5), 1–12 (2019)
40. Wu, C., Bi, X., Pfrommer, J., Cebulla, A., Mangold, S., Beyerer, J.: Sim2real transfer learning for point cloud segmentation: An industrial application case on autonomous disassembly. In: *WACV*. pp. 4531–4540 (2023)
41. Wu, C., Fu, H., Kaiser, J.P., Barczak, E.T., Pfrommer, J., Lanza, G., Heizmann, M., Beyerer, J.: 6d pose estimation on point cloud data through prior knowledge integration: A case study in autonomous disassembly. *Procedia CIRP* **122**, 193–198 (2024)
42. Wu, C., Huang, Q., Jin, K., Pfrommer, J., Beyerer, J.: A cross branch fusion-based contrastive learning framework for point cloud self-supervised learning. In: *3DV*. pp. 528–538 (2024)
43. Wu, C., Zheng, J., Pfrommer, J., Beyerer, J.: Attention-based point cloud edge sampling. In: *CVPR*. pp. 5333–5343 (2023)
44. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: *ICCV*. pp. 10033–10041 (2021)
45. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS* **35**, 33330–33342 (2022)

46. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D ShapeNets: A deep representation for volumetric shapes. In: CVPR. pp. 1912–1920 (2015)
47. Xia, Y., Gladkova, M., Wang, R., Li, Q., Stilla, U., Henriques, J.F., Cremers, D.: Casspr: Cross attention single scan place recognition. In: ICCV. pp. 8461–8472 (2023)
48. Xu, C., Zhai, B., Wu, B., Li, T., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: You only group once: Efficient point-cloud processing with token representation and relation inference module. In: IROS. pp. 4589–4596. IEEE (2021)
49. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics* **35**(6), 1–12 (2016)
50. Zhang, C., Wan, H., Shen, X., Wu, Z.: Patchformer: An efficient point transformer with patch attention. In: CVPR. pp. 11799–11808 (2022)
51. Zhang, Z., Hua, B.S., Rosen, D.W., Yeung, S.K.: Rotation invariant convolutions for 3d point clouds deep learning. In: 3DV. pp. 204–213 (2019)
52. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: CVPR. pp. 10076–10085 (2020)
53. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV. pp. 16259–16268 (2021)
54. Zhong, Q., Han, X.F.: Point cloud learning with transformer. arXiv preprint [arXiv:2104.13636](https://arxiv.org/abs/2104.13636) (2021)
55. Zhou, C., Luo, Z., Luo, Y., Liu, T., Pan, L., Cai, Z., Zhao, H., Lu, S.: PTTR: Relational 3d point cloud object tracking with transformer. In: CVPR. pp. 8531–8540 (2022)
56. Zhou, H., Cao, Y., Chu, W., Zhu, J., Lu, T., Tai, Y., Wang, C.: SeedFormer: Patch seeds based point cloud completion with upsample transformer. In: ECCV. pp. 416–432 (2022)



A Hyperparameter Optimization Method Based on Statistical Orthogonal Design for Neural Network Models

Yu Wang^{1,3(✉)}, Bo Du², Shufan Wu², and Xingli Yang^{2,3}

¹ School of Modern Educational Technology, Shanxi University, Taiyuan 030006, China

wangyu@sxu.edu.cn

² School of Mathematics and Statistics, Shanxi University, Taiyuan 030006, China

³ Key Laboratory of Complex Systems and Data Science, Shanxi University, Ministry of Education, Taiyuan 030006, China

Abstract. In neural network models, hyperparameters have a significant impact on model performance. Currently, the commonly used hyperparameter optimization methods include manual search, grid search, random search, Bayesian optimization, and so on. However, these methods always exhibit some problems such as high computational cost, low convergence rate and poor model performance. Thus, for image classification tasks, a hyperparameter optimization method based on statistical orthogonal design for neural network models is proposed in this paper. With the same number of experiments, the classification accuracy of the proposed method is significantly better than that of grid search, random search, and Bayesian optimization methods for both two-level and three-level orthogonal designs. With the same classification accuracy as grid search, random search, and Bayesian optimization methods, the proposed method has fewer experimental times. Furthermore, the single-factor rotation method and statistical variance analysis technique are also applied to study the effect of different hyperparameters on the performances of the neural network models.

Keywords: Hyperparameter optimization · Neural network · Orthogonal design · Random search · Grid search · Bayesian optimization.

1 Introduction

In machine learning research, neural network models have become a benchmark model for classification, segmentation, detection and retrieval tasks [1], and it is

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.

widely used in unmanned driving [2], intelligent transportation [3,4] and smart healthcare [5]. Neural network model is a mathematical model for the nonlinear representation and logical operations on complex information, with the prototype of biological nervous system and the theoretical basis of network topology [1,2]. As the increased of layers and the number of neurons from a few to thousands, the size of neural network model is significantly increased. This also brings a sharp rise in the number of parameters. Generally, the parameters of neural network models include ordinary parameters and hyperparameters [6]. The ordinary parameters can be automatically learned from the sample data to achieve the optimal value, while the hyperparameters usually need to be set manually. Currently, it has been showed that hyperparameters have a very significant impact on the performance of neural network model. For example, the hyperparameter of learning rate has a significant impact on the update speed in the direction of the gradient. If the learning rate is small, it slows down the parameter update speed, leading to increased computational costs. Conversely, if the learning rate is large, it easily crosses the local minimum, and fails to converge. Therefore, how to select appropriate hyperparameters to optimize model performance, known as the hyperparameter optimization problem, is a great challenge in the neural network model research [7].

Hyperparameter optimization for neural network models is to select the best hyperparameters by minimizing the difference between actual labels and the predictions of the neural network models [8]. Many hyperparameter optimization strategies with different research backgrounds are proposed, such as manual search, grid search, random search, Bayesian optimization, and evolutionary algorithms. For instance, in the traditional three-layer feedforward neural network models with few hyperparameters, manual search is always used to find the optimal hyperparameters [9]. Unfortunately, manual search relies on experts' experience and intuition, and it might not always find the optimal solution.

To address the drawbacks of manual search, the grid search method for hyperparameter optimization is introduced. As an exhaustive search technique, grid search selects the optimal combination of hyperparameters by searching all possible combinations of hyperparameters within a predefined hyperparameter subset [10,11]. However, in the case of high dimensions, "dimensional disaster" is easily caused with the exponentially increased of the combinations of hyperparameters in grid search.

To overcome the high computational cost and optimization difficulties, [12] introduced an automatic random search optimization method and proved its superiority over manual search and grid search methods in theory. Random search has become a benchmark algorithm for hyperparameter optimization by randomly sampling from all the combinations of hyperparameter to greatly save search time and computational cost [13,14]. For example, [13] proposed an improved random search algorithm with an early-stopping mechanism and weight sharing, and achieved competitive neural network architectures in the neural architecture search task. Although random search has greatly improved the speed of hyperparameter optimization, it still needs a large number of exper-

iments to find the optimal solution due to the lack of directional guidance in the search process.

On the other hand, [15] applied the Bayesian non-parametric optimization method to the hyperparameter selection of the random forest model for improving the classification performance. Bayesian optimization [16, 17] first fits an unknown target function by using prior distribution, then it selects the next hyperparameter combination by the posterior distribution until optimal. The related study can also be found in [18–20]. In practice, Bayesian optimization is always used for scenarios with low dimensions, typically 10 to 20, and it is not suitable for discrete spaces and cannot be processed in parallel. In addition, some other hyperparameter optimization methods are also proposed. See [21, 22].

In summary, the above mentioned hyperparameter optimization methods such as manual search, grid search, random search, and Bayesian optimization, either require the abundant manual parameter optimization and trial-and-error or entail large computational costs. Thus, a hyperparameter optimization method based on statistical orthogonal design for neural network models is proposed, which has the characteristics of efficiency, speed, and cost-effectiveness. Experimental results on multiple neural network models show that the proposed method outperform grid search, random search and Bayesian optimization methods.

2 A Hyperparameter Optimization Method Based on Statistical Orthogonal Design for Neural Network Models

2.1 Hyperparameter Optimization

Hyperparameter optimization is to find the optimal hyperparameter $\lambda^{(*)}$ by minimizing the following objective function:

$$\lambda^{(*)} \approx \arg \min_{\lambda \in \Lambda} \sum_{Z_i \in D^{(valid)}} L\left(Z_i; A_\lambda\left(D^{(train)}\right)\right) \quad (1)$$

where, L is the loss function representing the difference between the true samples and the model predictions, $D^{(train)}$ and $D^{(valid)}$ refer to the training and validation sets respectively, and A_λ is the neural network model, which can be a fully connected neural network, deep belief network, convolutional neural network, etc.

Currently, grid search, random search and Bayesian optimization methods have been widely used in the hyperparameter optimization for neural network models. Then these three methods are respectively described. Followed, the proposed hyperparameter optimization method based on statistical orthogonal design for neural network models is also introduced.

2.2 Grid Search

As an exhaustive search technique, grid search selects the optimal combination of hyperparameters by searching all possible combinations of hyperparameters within a predefined hyperparameter subset. For example, in a grid search with three factors, there are three hyperparameters to be optimized: α , β , and γ . Here, α takes values from $\{1, 2, 3\}$, β takes values from $\{0.1, 0.2, 0.3\}$, and γ takes values from $\{True, False\}$. Then the grid search method is to select the optimal combination of hyperparameters by searching all 18 hyperparameter combinations.

2.3 Random search

Random search is to find the optimal values of hyperparameters by using the random sampling technology. It is firstly assumed that each hyperparameter follows a marginal distribution, such as Bernoulli distribution, log-uniform distribution, or normal distribution, and then the hyperparameter search is carried out by random sampling.

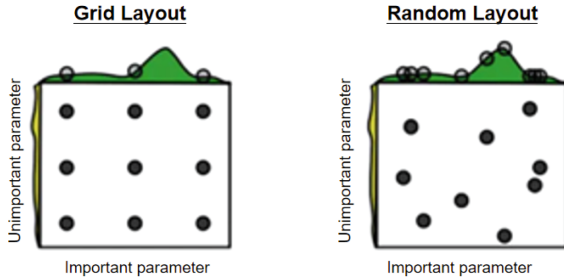


Fig. 1. Grid and random search of nine trials for optimizing a function $f(x, y) = g(x) + h(y) \approx g(x)$ with low effective dimensionality (Fig. 1 is from reference [12]).

Different from grid search, random search does not need to discretize the hyperparameter value interval, but randomly sample in the hyperparameter value interval, which ensures that random search can be optimized in a larger hyperparameter space. As shown in Fig. 1, with grid search, nine trials only test $g(x)$ in three distinct places, however, with random search, all nine trials explore distinct values of $g(x)$.

2.4 Bayesian Optimization

Bayesian optimization is an iterative hyperparameter selection method based on probability distribution. It first uses the probabilistic agent model defined by a Gaussian process to model the objective function, and then uses the collection function to select the next evaluation point. After the evaluation point is assessed, the model is updated with the new evaluation data.

Table 1. Seven-factor two-level orthogonal array

	A	B	C	D	E	F	G
1	1	2	1	2	1	2	2
2	2	1	2	1	1	2	2
3	1	2	2	1	2	2	1
4	1	1	2	2	2	1	2
5	2	2	2	2	1	1	1
6	2	1	1	2	2	2	1
7	1	1	1	1	1	1	1
8	2	2	1	1	2	1	2

2.5 A Hyperparameter Optimization Method Based on Statistical Orthogonal Design for Neural Network Models

Orthogonal design is a reasonable arrangement and design method for multi-factor and multi-level experimental points. It only needs to extract some representative points from the full experiment, and then arrange and design the experiment according to the orthogonal property, it can achieve the predetermined goal with less experiment times and greatly reduce the experiment cost [23, 24]. For example, in an experiment with seven factors, each having two levels, a full factorial design would require $2^7 = 128$ experimental setups. However, if we use the orthogonal design, only eight experiments are needed by selecting a suitable orthogonal array for seven factors with two levels (as shown in Table 1, where rows represent experiment numbers, columns represent factors, and ‘1’ and ‘2’ indicate the levels of factors). The experimental result with eight experiments is close to those of 128 experiments, and the experimental cost is only 1/16 of that of the full experiment.

In view of the excellent properties of statistical orthogonal design, this paper integrates the concept of orthogonal design into the hyperparameter optimization of neural network models, and proposes a hyperparameter optimization method based on the orthogonal design. The algorithm process is as follows: First Step: Select the hyperparameters to be optimized and determine their value ranges; Second Step: Divide the value ranges of the hyperparameters into multiple intervals, and discretize the continuous hyperparameters to define levels of values, known as the levels of hyperparameters. Next, choose an appropriate orthogonal array based on the number and levels of hyperparameters to be optimized; Third Step: Combine the levels of all hyperparameters to be optimized according to the orthogonal table, and select the optimal combination of hyperparameter levels by calculating the scores of all combinations; Fourth Step: For the value range of each hyperparameter in the optimal combination obtained in the previous step, repeat the selection process for the optimal combination of hyperparameter levels until further division and discretization of the hyperparameter value range

Algorithm 1: Hyperparameter Optimization Based on Orthogonal Design

Input: Hyperparameters $\Lambda = \{\lambda_j, j = 1, 2, \dots, n\}$, where $\lambda_j \in (a_j^{(0)}, b_j^{(0)})$, Number of splits S

Output: Optimal combination of hyperparameters Λ_*

```

1: for  $k = 1; k \leq S; k++$  do
2:   Divide the range of each hyperparameter into:
3:    $I_{j1}^{(k)} = (a_{j1}^{(k)}, b_{j1}^{(k)}), I_{j2}^{(k)} = (a_{j2}^{(k)}, b_{j2}^{(k)})$ , where  $b_{j1}^{(k)} = a_{j2}^{(k)}$ ;
4:   Discretize to obtain:  $\lambda_{j1}^{(k)} \in I_{j1}^{(k)}, \lambda_{j2}^{(k)} \in I_{j2}^{(k)}$ ;
5:   Select an orthogonal array  $OA(m, n) = (\alpha_{ij}), i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ;
6:   for  $i = 1; i \leq m; i++$  do
7:     for  $j = 1; j \leq n; j++$  do
8:       if  $\alpha_{ij} = '1'$  then
9:          $\lambda_{ij}^{(k)} = \lambda_{j1}^{(k)}$ ;
10:        else
11:          $\lambda_{ij}^{(k)} = \lambda_{j2}^{(k)}$ ;
12:        end if
13:      end for
14:    end for
15:    Obtain  $m$  hyperparameters:  $\Lambda^{(k)} = \{\Lambda_1^{(k)}, \dots, \Lambda_m^{(k)}\}$ ;
16:    Calculate scores to find the highest scoring hyperparameters
17:     $\Lambda_*^{(k)} = \arg \max_{\Lambda_j^{(k)} \in \Lambda^{(k)}} A(\Lambda_j^{(k)})$ ;
18:  end for

```

is not possible. The final combination of hyperparameter levels obtained at this point is considered the optimal set of hyperparameters.

Specifically, Algorithm 1 shows the hyperparameter selection process of the proposed method for the two-level orthogonal design. For the three-level orthogonal design, we only need to discretize the range of hyperparameter values into three parts in Step 2 and obtain the three levels of hyperparameter values. At the same time, in Step 4, a three-level orthogonal array should be selected.

3 Experimental Analysis

To verify the effectiveness and superiority of the proposed hyperparameter optimization method based on statistical orthogonal design for neural network models, this paper compares its performance with traditional grid search, random search, and Bayesian optimization. In addition, we also analyze the impact of different hyperparameters on the neural network model performance.

3.1 Experimental Setup and Procedure

Four popular neural network models of three-layer fully connected neural network, four-layer deep belief network, three-layer convolutional neural network,

and VGGNet11 network, and three image classification datasets of MNIST, Rectangle, and Cifar10, are selected to compare the performance of the four hyperparameter optimization methods. The experimental setup of each neural network model is described below.

Three-layer Fully Connected Neural Network (FCN): The three-layer fully connected neural network model is composed of input layer, hidden layer and output layer. The number of neurons in the input layer is 784, and in the output layer is 10. The model is comprised of 5 hyperparameters. The value range of each hyperparameter is shown in Table 2.

Specifically, for two-level orthogonal design, the value range of the hyperparameter was segmented 6 times, and 8 groups of hyperparameters were tested each time. Therefore, $6 \times 8 = 48$ experimental results were obtained in this repetition, and the average was calculated as the result of this repetition. To avoid random occurrences and ensure reliable results, a total of 50 repeated experiments were conducted. Similar to the two-level orthogonal design, the hyperparameter optimization based on the three-level orthogonal design was also carried out 50 repeated experiments.

Table 2. Value range of hyperparameters

Model	Hyperparameter	Range	Hyperparameter	Range
FCN	Learning Rate	0.001–5	Batch Size	20–100
	Simulated Annealing	100–10000	Hidden Layer Nodes	16–1024
	L_2 Regularization Coefficient	$3.1e^{-7} - 3.1e^{-5}$		
DBN	Global Learning Rate	0.001–0.1	Global Simulated Annealing	10–400
	Local Learning Rate	0.001–0.1	Local Rounds	10–200
	Local Simulated Annealing	128–512	Local Hidden Layer Nodes	10–400
CNN	Learning Rate	$10e^{-5} - 10e^{-3}$	Batch Size	32–128
	Weight Distribution	0.1–0.3	Convolutional Kernel Size	1, 3, 5, 7
	Number of Convolutional Kernels	32–128	Fully Connected Layer Nodes	256–1024
VGG	Learning Rate	$10e^{-4} - 10e^{-2}$	Batch Size	16–64
	Convolutional Kernel Size	1,3 ... 13, 15	Number of Convolutional Kernels	32–512
	Weight Distribution	0.01–0.03		

Four-Layer Deep Belief Network (DBN): The four-layer deep belief network model is composed of three stacked restricted boltzmann machines [25, 26]. Because deep belief network is pre-trained layer by layer before global fine-tuning, the hyperparameters can be divided into local hyperparameters and global hyperparameters. In this experiment, 2 global hyperparameters and 12 local hyperparameters are to be optimized, and the value range of hyperparameters is shown in Table 2.

When a two-level orthogonal design is used to optimize the four-layer deep belief network, a fourteen-factor two-level orthogonal array (16 rows by 14 columns) was selected, resulting in 16 hyperparameter combinations. For the three-level orthogonal design, a fourteen-factor three-level orthogonal array (54 rows by 14 columns) was selected. The two-level and three-level orthogonal design optimization experiments were all repeated 20 times.

Three-Layer Convolutional Neural Network (CNN): The three-layer convolutional neural network model is comprised of two convolutional layers and a fully connected layer, and each convolutional layer is connected with a pooling layer. The model is comprised of 9 hyperparameters. For each convolutional layer, three hyperparameters are involved: the size of the convolutional kernel, the number of convolutional kernels, and the distribution of weights on the convolutional kernel. In addition, three hyperparameters are considered: batch size, learning rate, and the number of neurons in the fully connected layer. The value range of hyperparameters is shown in Table 2.

The number of repeated experiments was set to 20. In the two-level orthogonal design optimization experiment, a nine-factor two-level orthogonal array (12 rows by 9 columns) was selected. And in the three-level orthogonal design experiment, a nine-factor three-level orthogonal array (27 rows by 9 columns) was selected.

VGGNet11 Network (VGG): The VGGNet11 model, with a more complex structure and more parameters, is comprised of 8 convolutional layers, 4 pooled layers and 3 fully connected layers, and includes 25 hyperparameters. In the two-level orthogonal design optimization experiment, a twenty-five-factor two-level orthogonal array (28 rows by 25 columns) was selected, and a total of 20 repetitions were performed.

3.2 Results and Analysis

Experimental Results and Analysis for the Three-Layer Fully Connected Neural Network Model Under the same experimental conditions, the method proposed in this paper was compared with grid search, random search, and Bayesian optimization methods on the MNIST and Rectangle datasets by using a three-layer fully connected neural network model. The experimental results are shown in the left figure of Fig. 2. In the figure, the green dashed line represents the average of all results obtained by grid search. The first and second columns represent the results of 48 experiments using random search and Bayesian optimization, respectively. The third, fourth, fifth, and sixth columns represent the results of 48, 40, 32, and 24 repeated experiments using two-level orthogonal design, respectively.

From Fig. 2, we can see that for the two-level and three-level orthogonal designs, the proposed method all can achieve higher classification accuracy with fewer experiments than grid search, random search, and Bayesian optimization

methods on the MNIST and Rectangle datasets. For example, in the experiment of hyperparameters optimized by two-level orthogonal design on MNIST dataset, the average classification accuracy of grid search is 90.77%, the median classification accuracy of 48 random searches is 93.08%, the median classification accuracy of 48 Bayesian optimizations is 93.20%, and the median obtained by searching 24 times using two-level orthogonal design is 93.14%, which is 2.37% higher than grid search, 0.06% higher than random search. When the number of experimental times increases from 24 to 32, 40, and 48, the median classification accuracies of the proposed method are 93.57%, 93.86%, and 94.15%, respectively, which are 2.80%, 3.09%, and 3.38% higher than grid search, 0.49%, 0.78%, and 1.07% higher than random search, and 0.37%, 0.66%, and 0.95% higher than Bayesian optimization, respectively. Actually, when the number of searches reaches 40 or more, the box of the two-level orthogonal design is significantly higher than that of random search and Bayesian optimization, and there is no intersection between the boxes. These all indicate that the proposed method is significantly better than random search and Bayesian optimization.

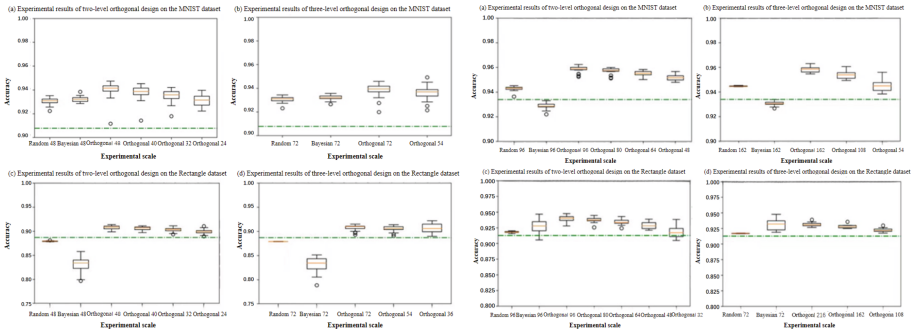


Fig. 2. Experimental results for the three-layer fully connected neural network and four-layer deep belief network models

Experimental Results and Analysis for the Four-Layer Deep Belief Network Model

Right figure of Fig. 2 shows the comparison results of the proposed method, grid search, random search, and Bayesian optimization methods for the four-layer deep belief network model on the same datasets. As shown in Fig. 2, the classification accuracies obtained by all hyperparameter optimization methods for the four-layer deep belief network are improved compared to the three-layer fully connected neural network. However, the proposed method still has the best performance. For example, on the Rectangle dataset, the classification accuracies of grid search, random search, Bayesian optimization, and the proposed method are 91.27%, 91.81%, 92.74%, and 94.02%, respectively, which increased by 3.97%, 3.86%, 9.59%, and 3.13% compared with the three-layer fully connected neural network. The box interval of 96 random searches

is $[0.9174, 0.9190]$, the box interval of 96 Bayesian optimizations is $[0.9202, 0.9347]$. However, the box interval of 96 searches based on two-level orthogonal design is $[0.9366, 0.9425]$, significantly higher than the boxes for random search and Bayesian optimization and with no crossover. And the proposed method improves by 1.88% compared with grid search and 1.38% compared with random search, but it has no significant difference from Bayesian optimization (Bayesian optimization method has large variance fluctuations).

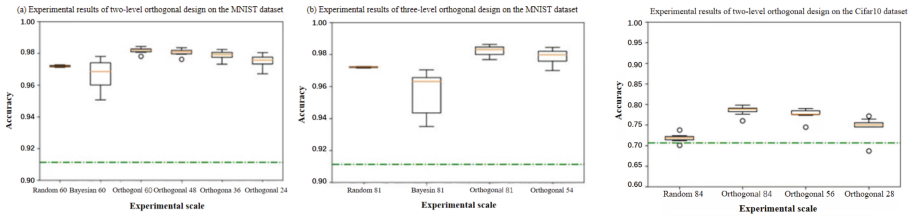


Fig. 3. Experimental results for the three-layer convolutional neural network and VGGNet11 network models

Experimental Results and Analysis for the Three-Layer Convolutional Neural Network Model When using a three-layer convolutional neural network to classify the MNIST dataset, the classification accuracies of all methods are further improved compared with the four-layer deep belief network. Experimental results are shown in the left figure of Fig. 3. For example, on the MNIST dataset, the accuracies of random search, Bayesian optimization, and the proposed method are 97.19%, 96.62%, and 98.19%, respectively, which are 2.91%, 3.77%, and 2.35% higher than that of the four-layer deep belief network. Even if the average classification accuracy reaches 97.20%, the method proposed in this paper can still improve by about one percentage point, and the number of experiments required is greatly reduced. Specifically, the results obtained by the proposed method after 24 searches are not significantly different from those obtained by random search for 60 times, and are better than those obtained by Bayesian search for 60 times. With the same number of experiments, the proposed method is overall 4.80% higher than grid search, 1.00% higher than random search, and 1.57% higher than Bayesian optimization.

Experimental Results and Analysis for the VGGNet11 Network Model In the optimization of the VGGNet11 network model with 25 hyperparameters, the proposed method still achieved significant results, and only 28 experiments could achieve better results than 84 random searches, saving 2/3 of the computing cost. Furthermore, with the increased optimization times of orthogonal design, the classification accuracy reached 77.60% after 56 searches,

which was 5.78% higher than random search. At the same 84 searches, the overall accuracy improved to 78.50%, which was 6.68% higher than random search and 7.85% higher than grid search, as shown in the right figure of Fig. 3.

In summary, the experiment results with multiple datasets and neural network models all demonstrate the effectiveness and superiority of the proposed method. With the same number of experiments, the classification accuracy of the proposed method is significantly better than that of grid search, random search, and Bayesian optimization methods for both two-level orthogonal design and three-level orthogonal design. With the same classification accuracy as the grid search, random search, and Bayesian optimization methods, the proposed method requires fewer experimental times.

3.3 Analysis of the Importance of Different Hyperparameters

Although the neural network model contains many hyperparameters, we find that different hyperparameters have different effects on the model performance. Some hyperparameters have a great impact on the model performance, while others have almost no impact on the model performance. Thus, in this section, the importance of hyperparameters is also analyzed by using the single-factor rotation method and statistical analysis of variance.

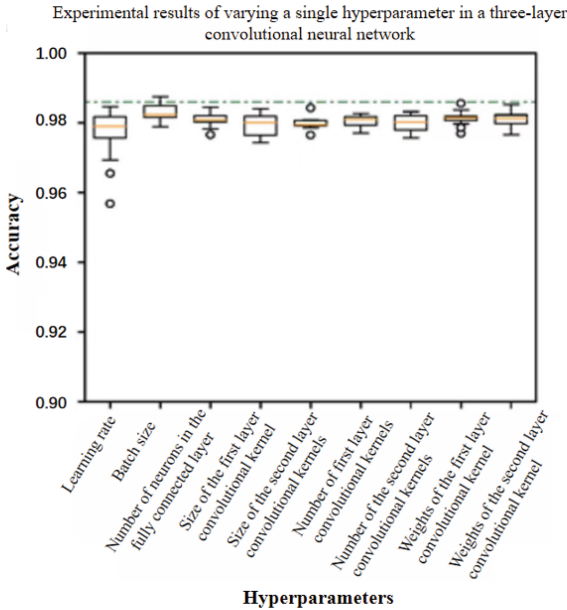


Fig. 4. Experimental results for the single-factor rotation method

First, the importance of 9 hyperparameters (learning rate, batch size, number of neurons in the fully connected layer, and the size, number, weight distribution

of the first-layer and second-layer convolutional kernels) in the three-layer convolutional neural network model are analyzed based on the single-factor rotation method. The experimental results are shown in Fig. 4. The figure reveals that different hyperparameters obviously have different influences on the classification accuracy of the model. For example, the fluctuation range of the learning rate hyperparameter with the largest fluctuation is $[0.9568, 0.9847]$, with a difference of 2.79%. In contrast, there is the least impact on the classification accuracy of the model by changing the number of convolutional kernels of the first layer, and the fluctuation range is $[0.9771, 0.9816]$, the difference is only 0.45%. For other hyperparameters such as learning rate, batch size, and the size of the first-layer convolutional kernels, their fluctuations all exceed 1%, indicating a significant impact on model performance. But the number of convolutional kernels in the first and second layers have a minimal effect on model performance, each less than 0.50%. In addition, we also conducted an importance analysis of hyperparameters based on statistical variance analysis techniques.

Table 3. Variance contribution of hyperparameters in three-layer convolutional neural network model

Hyperparameter	p-value	Variance Contribution
Learning Rate	$2.2e^{-16}$	0.8700
Number of Second-Layer Convolutional Kernels	$8.9e^{-14}$	0.0620
Number of Neurons in Fully Connected Layer	$7.1e^{-7}$	0.0272
Batch Size	$1.1e^{-6}$	0.0263
Size of First-Layer Convolutional Kernels	0.0143	0.0066
Size of Second-Layer Convolutional Kernels	0.0624	0.0038
Weight of First-Layer Convolutional Kernels	0.1292	0.0025
Weight of Second-Layer Convolutional Kernels	0.2400	0.0015
Number of First-Layer Convolutional Kernels	0.8608	0.00003

Specifically, 1794 samples generated by randomly sampling from 9 hyperparameters of three-layer convolutional neural network model are used to fit a linear regression model:

$$y = 0.148x_1 + 0.038x_2 + 0.025x_3 + 0.022x_4 + 0.018x_5 + 0.016x_6 - 0.0078x_7 - 0.0055x_8 - 0.00079x_9 + 0.8515 \quad (2)$$

where, x_1, x_2, \dots, x_9 represent the hyperparameters of learning rate, number of second-layer convolutional kernels, number of fully connected neurons, batch size, size of first-layer convolutional kernels, size of second-layer convolutional kernels, weight of first-layer convolutional kernels, weight of second-layer convolutional kernels, and number of first-layer convolutional kernels, respectively. In the significance test of the regression model with a significance level of $\alpha = 0.05$,

the calculated F statistic is $113.92 > F_{9,1784}(0.05) = 1.94$, leading to the rejection of the null hypothesis. This means that hyperparameters have a significant impact on the classification performance of the model. Then, we conducted the significance tests of the regression coefficients, and the experimental results are shown in Table 3. From Table 3, we can see that 5 hyperparameters are significant, namely learning rate, number of second-layer convolutional kernels, number of fully connected layers, batch size, and size of first-layer convolutional kernels; while the remaining 4 hyperparameters have no significant impact on the results.

Moreover, the variance contribution rate is also considered to measure the importance of hyperparameters. As shown in Table 3, the hyperparameter of learning rate has the largest variance contribution rate, reaching 87%; the second largest hyperparameter is the number of second-layer convolutional kernels, with a variance contribution rate of 6.20%; while the variance contribution rate of the number of first-layer convolutional kernels is the lowest, less than one ten-thousandth. Overall, the cumulative variance contribution of the first five hyperparameters reaches 99.21%, while the variance proportion of the last four hyperparameters is only 0.79%.

Finally, Fig. 5 shows the quantitative comparison results of the impact of significant and non-significant hyperparameter subsets on the classification results of the CNN model. As we can see, in 100 repeated experiments, the subset of significant hyperparameters represented by the first column has a large fluctuation in model classification accuracy, with a range of $[0.9115, 0.9863]$ and a width of 0.0748. In the second column, the fluctuation range corresponding to the nonsignificant hyperparameter subset is $[0.9827, 0.9898]$, the width is 0.0071, and the fluctuation is about 1/10 of the significant subset. These results all indicate that the subset of significant hyperparameters has an important impact on

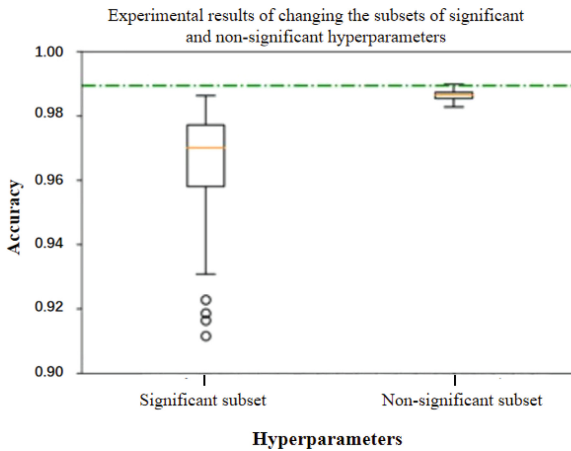


Fig. 5. Impact of significant and non-significant hyperparameter subsets on the classification results of the CNN model

model performance, while the subset of nonsignificant hyperparameters has a minimal impact.

4 Conclusion and Future Work

In this paper, a neural network hyperparameter optimization method based on statistical orthogonal design for image classification problems is proposed. Compared to traditional grid search, random search, and Bayesian optimization methods, the proposed method has the following advantages: First, the value space of each hyperparameter is segmented continuously until the hyperparameter converges to the best range. And the whole optimization process has a clear directionality, which can effectively alleviate or solve the disorder of the random search method and the instability of the Bayesian optimization method. Second, in the process of hyperparameter optimization, the optimal hyperparameter is only searched in a partial value ranges of hyperparameters, which can greatly save the computational overhead caused by repeated values on unimportant hyperparameters in grid search. Moreover, the experimental results show that the proposed method improves classification accuracy by 0.90%, 1.46%, and 1.02% over random search for three-layer fully connected neural networks, four-layer deep belief networks, and three-layer convolutional neural networks on MNIST dataset, and by 3.18%, 2.44%, and 7.09% over grid search, and 0.87%, 3.04%, and 1.57% over Bayesian optimization. On Rectangle dataset, the overall improvements compared to random search methods are 2.86% and 1.79%, for three-layer fully connected neural networks and four-layer deep belief networks. Compared to grid search methods, the improvements are 2.05% and 2.29%, and compared to Bayesian optimization methods, the improvements are 7.61% and 1.23%, respectively. In the Cifar10 dataset classification tasks based on VGGNet11 network, the improvement of 6.68% over random search and 7.85% over grid search is achieved. Furthermore, the importance of hyperparameters is also analyzed by the single-factor rotation method and statistical analysis of variance.

In the future, we will consider applying the proposed method to the image segmentation and object detection tasks to further verify its effectiveness and universality.

Acknowledgements. The experiments are supported by High Performance Computing System of Shanxi University. Yu Wang is the corresponding author. Bo Du and Shufan Wu contributed equally to this work.

References





1. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015)
2. Stilgoe, J.: Machine learning, social learning and the governance of self-driving cars. *Soc. Stud. Sci.* **48**, 25–56 (2018)

3. Ruan, W., Jianfa, W.U., Kou, Z., et al.: Obstacle visual sensing based on deep learning for low-altitude small unmanned aerial vehicles. *Scientia Sinica Informationis* **50**, 692–703 (2020)
4. Polson, N.G., Sokolov, V.O.: Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies* **79**, 1–17 (2017)
5. Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., et al.: Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, 20170387 (2018)
6. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**, 436–444 (2015)
7. Glorot X., Bengio Y.: Understanding the difficulty of training deep feedforward neural networks. In: 13th International Conference on Artificial Intelligence and Statistics, 249–256 (2010)
8. Yu T., Zhu H.: Hyper-parameter optimization: a review of algorithms and applications. arXiv Preprint [ArXiv:2003.05689](https://arxiv.org/abs/2003.05689) (2020)
9. Zhongjin, Y.: Architecture optimization for neural networks. *Comput. Eng. Appl.* **25**, 52–54 (2004)
10. Larochelle H., Erhan D., Courville A., et al.: An empirical evaluation of deep architectures on problems with many factors of variation. In: 24th International Conference on Machine Learning, 473–480 (2007)
11. Lerman, P.M.: Fitting segmented regression models by grid search. *J. Roy. Stat. Soc.* **29**, 77–84 (1980)
12. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13**, 281–350 (2012)
13. Li L., Talwalkar A.: Random search and reproducibility for neural architecture search. In: *Uncertainty in Artificial Intelligence*, 367–377 (2020)
14. Liashchynskyi P., Liashchynskyi P.: Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv Preprint [ArXiv:1912.06059](https://arxiv.org/abs/1912.06059) (2019)
15. Hutter F., Hoos H.H., Leyton-Brown K.: Sequential model-based optimization for general algorithm configuration. In: *International Conference on Learning and Intelligent Optimization*, 507–523 (2011)
16. Shahriari, B., Swersky, K., Ziyu, W., et al.: Taking the human out of the loop: a review of Bayesian optimization. *Proc. IEEE* **104**, 148–175 (2015)
17. Bergstra J., Bardenet R., Bengio Y., et al.: Algorithms for hyper-parameter optimization. In: *24th International Conference on Neural Information Processing Systems*, 2546–2554 (2011)
18. Wang J., Xu J., Wang X.: Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning. arXiv Preprint [ArXiv:1801.01596](https://arxiv.org/abs/1801.01596) (2018)
19. Liang, X.: Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering* **34**, 415–430 (2019)
20. Vincent, A.M., Jidesh, P.: An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms. *Sci. Rep.* **13**, 4737 (2023)
21. Boulesteix, A.L., Bischl, B., Deng, D., et al.: Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **13**, e1484 (2023)
22. Chen, S.P., Wu, J., Liu, X.Y.: EMORL: effective multi-objective reinforcement learning method for hyperparameter optimization. *Eng. Appl. Artif. Intell.* **104**, 104315 (2021)
23. Mandenius, C.F., Brundin, A.: Bioprocess optimization using design-of-experiments methodology. *Biotechnol. Prog.* **24**, 1191–1203 (2008)

24. Weissman S.A., Anderson N.G.: Design of experiments (DoE) and process optimization. A review of recent publications. *Organic Process Research and Development* **19**, 1605-1633 (2015)
25. Hinton G.E.: A practical guide to training restricted Boltzmann machines. In: *Neural networks: Tricks of the trade*, 599-619 (2012)
26. Erhan D., Bengio Y., Courville A., et al.: Why does unsupervised pre-training help deep learning? In: *13th International Conference on Artificial Intelligence and Statistics*, 201-208 (2010)



NeuroDAVIS-FS: Feature Selection Through Visualization Using NeuroDAVIS

Chayan Maitra¹ , Anwesha Sengupta² , and Rajat K. De¹  

¹ Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India
rajat@isical.ac.in

² Department of Applied Statistics, Maulana Abul Kalam Azad University of Technology, Haringhata, West Bengal, India

Abstract. Dealing with high-dimensional datasets is challenging nowadays due to computational complexity, the curse of dimensionality, and model overfitting. It becomes necessary to reduce the dimension of the dataset for a better understanding of inherent information. Feature selection techniques are widely utilized to rank features based on their importance and accordingly reduce the dimension of the original datasets with respect to this ranking. Existing feature selection methods are mainly developed for specific downstream tasks and show several drawbacks e.g., not considering the inherent associations and their importance. In most of the cases, the methods are computationally expensive as well. In order to address such drawbacks, the present study aims to propose a feature selection method, called NeuroDAVIS-FS, which performs in an unsupervised learning setup without assuming any prior data distribution. Initially, it considers training using the model NeuroDAVIS, developed earlier for data visualization, and selects features according to the trained model. The efficacy of the proposed NeuroDAVIS-FS has been demonstrated on various datasets from different domains and found to be effective in comparison with state-of-the-art feature selection methods. In addition, two case studies on image and biological datasets with a very low sample-feature ratio, have been executed and found to be effective for relevant feature selection.

Keywords: Dimensionality Reduction · Feature Selection · Feature Extraction · Data Visualization.

1 Introduction

In the rapid advancement of data science, the availability of high-dimensional datasets poses significant challenges like computational complexity, curse of dimensionality, and model overfitting. To deal with such high-dimensional datasets, dimensionality reduction is widely used for data pre-processing which in turn is effective for further data analysis [8]. Existing literature often follows two different types of dimension reduction approaches, viz., feature extraction and

feature selection [12]. Feature extraction techniques such as Principal Component Analysis (PCA), Auto-encoders, t-distributed Stochastic Neighbour Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP), etc., are used to generate new sets of features that preserve relevant information of the original dataset [2]. However, these feature extraction techniques cannot recognize the proper weightage of the actual features, which sometimes becomes less relevant to some specific tasks [9]. At the same time, feature selection techniques can address the issues by ranking the features based on their importance and accordingly reduce the dimension of the original datasets with respect to this ranking [9]. Existing state-of-the-art feature selection methods have some major drawbacks, e.g., huge space and time complexity, usage of the target class, and ignoring the inherent association among the features while ranking them, etc [4–6, 13, 15]. To address these limitations (Table 1), the present study aims to propose a novel and effective approach, called NeuroDAVIS-FS (*NDFS*), which is an extension of NeuroDAVIS, a method for data visualization [11].

Researchers have proposed a wide variation of feature selection methods based on specific downstream tasks. The feature selection methods are mainly classified in 3 broad categories, viz., filter method, wrapper method, and embedded method [3, 18]. Each of the techniques has its characteristics, advantages, and disadvantages e.g., the filter method utilizes statistical approaches to rank the features. Variance Threshold (*VT*), Chi-Squared test, ANOVA (Analysis of Variance), and Correlation coefficient are examples of filter methods. The major drawback of the filter method is that it assumes that the features are independent and does not consider the inherent association among the feature variables. Moreover, it does not take into account the features' importance. The wrapper methods evaluate the importance of the feature on the basis of the model's performance and accordingly select features. The corresponding examples are Forward Selection, Backward Elimination, and Recursive Feature Elimination (*RFE*). However, these methods are computationally expensive and sensitive to the specific model. On the other hand, the embedded method uses ensemble models for the same. Lasso, Random Forest, and Gradient Boosting Machines are examples of such feature selection methods. These are again computationally complex and prone to model overfitting. In view of the existing approaches, researchers have explored some novel approaches that differ from classical techniques. Liu and Zheng have proposed a novel feature selection method e.g., filtered and supported sequential forward search (*FS_SFS*) to enhance the performance of the Support Vector Machine classifier [10].

To address the shortcomings of classical feature selection techniques, a novel feature selection method, NeuroDAVIS-FS (*NDFS*), has been proposed in this study, which is an extension of the data visualization model, NeuroDAVIS. *NDFS* performs in an unsupervised learning setup without assuming any prior data distribution. Initially, it considers training using the model NeuroDAVIS and selects features according to the trained model. The performance has been demonstrated on three datasets, viz., *Breast Cancer*, *Wine*, and *Digits*, and compared against the state-of-the-art methods, viz., Variance Threshold (*VT*), GenericUnivariate-

Table 1. A theoretical comparison of existing state-of-the-art and proposed feature selection methods

Method	Technique	Advantages	Drawbacks
<i>VT</i> [6]	Unsupervised	<ul style="list-style-type: none"> Removes features with variance lower than certain threshold Simple and efficient Removes noise 	<ul style="list-style-type: none"> Ignores the inherent association among the features Threshold parameter is sensitive Does not consider Feature importance
<i>GUS</i> [13]	Supervised	<ul style="list-style-type: none"> Considers univariate statistical tests. Simple and efficient Selects target specific features. 	<ul style="list-style-type: none"> Limited to Univariate statistical tests Often overfits Ignores the inherent association among the features.
<i>SKB</i> [5]	Supervised	<ul style="list-style-type: none"> Simple and efficient Handles both categorical and numerical data Removes noise 	<ul style="list-style-type: none"> Ignores the inherent association among the features Sensitive to threshold parameter Ignores feature importance
<i>MIC</i> [4]	Supervised	<ul style="list-style-type: none"> Captures non-linear hidden patterns Handles both categorical and numerical data Robust in nature 	<ul style="list-style-type: none"> Computationally expensive Sensitive to noise Ignores feature-feature associations
<i>RFE</i> [5]	Supervised	<ul style="list-style-type: none"> Considers association among features Automatically ranks features based on their importance Reduces overfitting 	<ul style="list-style-type: none"> Huge computational cost Sensitive to model Requires careful cross-validation
<i>RFECV</i> [15]	Supervised	<ul style="list-style-type: none"> Considers association among features Automatically ranks features based on their importance Cross-Validation ensures generalized feature selection 	<ul style="list-style-type: none"> Huge computational cost Depends on the initial model Iterative process and cross-validation makes it even more complex
Proposed model (<i>NDFS</i>)	Unsupervised	<ul style="list-style-type: none"> Considers association among features Do not assume any inherent data distribution Effective for datasets with a very low sample-feature ratio 	<ul style="list-style-type: none"> Depends on the initial model Cannot handle categorical datasets directly

Select (*GUS*), SelectKBest (*SKB*), Mutual Information Classifier (*MIC*), Recursive Feature Elimination (*RFE*), and Recursive Feature Elimination with cross-validation (*RFECV*). Both the extracted and selected features have been compared with the state-of-the-art in terms of classification and clustering performances. In this context, Accuracy, Precision, Recall, and F1-score, are calculated for classification performances, on the other hand, Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Fowlkes-Mallows index (FMI) have been reported to evaluate the clustering performance respectively. Through this extensive study, it has been observed that the proposed model is robust enough to capture the hidden patterns present in datasets from various fields. In addition, two case studies on image and biological datasets with a very low sample-feature ratio (nearly 1 : 10), have been executed.

The outline of the present study is described as follows. Section 2 shows a detailed explanation of the proposed approach along with a problem scenario. Section 3 executes the outcome and analyses of the results. Section 4 draws the conclusion along with a direction for the extension of this study.

2 Problem statement and proposed approach

This section deals with the problem scenario along with the proposed approach.

2.1 Problem scenario

Dimension reduction plays a crucial role in data science as the number of features increases rapidly. Feature extraction methods can reduce the dimension of any dataset by producing a linear or non-linear combination. However, the extracted features are not at all interpretable. In this regard, feature selection methods are one step ahead of the feature extraction method and select relevant features by utilizing the domain knowledge. Motivated by this scenario, a feature selection module, *NDFS* has been proposed in this study, which is an extension of NeuroDAVIS, a data visualization approach.

Algorithm 1 NeuroDAVIS-FS

Input: A dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n = \{\mathbf{f}_j\}_{j=1}^d$ with n samples and d features.

Output: Top k features of \mathbf{X} , where $k \leq d$.

step 1: $Score \leftarrow []$

step 2: Drop columns with $std < 0.0001$.

step 3: $\mathbf{X}_{scaled} \leftarrow Minmaxscaling(\mathbf{X})$

step 4: Apply NeuroDAVIS. $\tilde{\mathbf{X}} \leftarrow NeuroDAVIS(\mathbf{X}_{scaled})$

step 5: Calculate $L_{KLD}(\mathbf{f}_j || \tilde{\mathbf{f}}_j), \forall j = 1, 2, \dots, d$ (using Eqn. (1)).

step 6: Calculate $Score_j, \forall j = 1, 2, \dots, d$ (using Eqn. (2)).

step 7: $F \leftarrow argmin_k Score$.

step 8: Return F

Let the dataset \mathbf{X} with n samples and d features be considered as input. The study aims to select the top k features based on the prior trained NeuroDAVIS model, where $k \leq d$. The proposed framework is described in Algorithm 1.

2.2 Proposed solution

This section discusses the description of the datasets that have been used to demonstrate the effectiveness of the proposed model along with the detailed methodology.

Data description In this study, various datasets from different domains have been used to demonstrate the effectiveness of the proposed feature selection module. The detailed descriptions are provided in Table 2. Among the mentioned datasets, *Breast Cancer*, *Wine* and *Digits* are utilized to evaluate the performance of the proposed model in comparison with the existing state-of-the-art methods. However, the remaining datasets are used to examine the performance of the proposed model when the sample-feature ratio is very low (nearly 1 : 10).

Table 2. Descriptions of the datasets utilized for the model evaluation

Name	Description	Samples	Features	Classes	Category & Source
<i>Breast Cancer</i>	Picture of a digitalized FNA of a breast mass is used to compute features.	569	30	2	Numeric [17]
<i>Wine</i>	A chemical examination of three distinct growers' wines, cultivated in Italy.	178	13	3	Numeric [1]
<i>Digits</i>	Grayscale images of handwritten digits	1797	64	10	Image [11]
<i>Coil20</i>	Grayscale images of objects, captured through different angles	1440	16384	20	Image [14]
<i>Olivetti Faces</i>	Grayscale images of human faces	400	4096	40	Image [16]
<i>Jurkat</i>	Sc-RNAseq data using the peripheral blood of a 14-year-old kid with acute lymphoblastic leukemia	3388	32738	10	Biological [19]

Methodology In this work, an extended version of NeuroDAVIS, viz., *NeuroDAVIS-FS*, has been proposed to select important features for data modeling and analysis [11]. Initially, NeuroDAVIS-FS considers the earlier data visualization module, NeuroDAVIS, for model training. In this context, NeuroDAVIS tries to reconstruct the data using random lower dimensional points. During the process of reconstruction, it learns suitable visualization of the dataset.

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n = \{\mathbf{f}_j\}_{j=1}^d$ denotes a dataset having n samples and characterized by d features. Where, \mathbf{x}_i and \mathbf{f}_j denote the i^{th} sample and j^{th} feature respectively. Initially, NeuroDAVIS takes an identity matrix of order n as input and creates random lower dimensional points at the *Latent layer*. These lower dimensional points are then projected to the original high-dimensional space at the *Reconstruction layer* through a few *Hidden layers*. This reconstructed data received from *Reconstruction layer*, has again been considered for calculating a loss and accordingly the weights and biases have been updated using Adam optimizer [11]. After successful training, NeuroDAVIS learns to recreate the original dataset and accordingly, 2 or 3 completely new features have been extracted at the *Latent layer* which are useful for visualization. It may be inferred that a feature with a lower variance contains less information and accordingly, it can easily be reconstructed by NeuroDAVIS.

Motivated by this aspect, a novel methodology, NeuroDAVIS-FS, has been developed in this module. In order to identify important features, we have examined how well the reconstructions are compared to the original dataset and how much variation is captured by that feature. Important features, that have been identified by NeuroDAVIS, have a high variance with a better reconstruction. After successful training, the reconstruction of the original features is represented by $\tilde{\mathbf{X}} = \{\tilde{\mathbf{f}}_j\}_{j=1}^d$. In order to validate the reconstruction capability, Kullback-Leibler Divergence (KLD) [7] has been used which is defined as

$$L_{KLD}(\mathbf{f}_j || \tilde{\mathbf{f}}_j) = \mathbf{f}_j \log\left(\frac{\mathbf{f}_j}{\tilde{\mathbf{f}}_j}\right) \quad (1)$$

For comparable results, initially, a preprocessing task i.e., Min-max scaling has been performed on each of the datasets and then fed to NeuroDAVIS. If NeuroDAVIS is able to reconstruct \mathbf{f}_j properly then $\tilde{\mathbf{f}}_j \sim \mathbf{f}_j$ and this indicates $L_{KLD}(\mathbf{f}_j || \tilde{\mathbf{f}}_j) \rightarrow 0$. Let $\sigma_{\mathbf{f}_j}$ be the standard deviation of the j^{th} feature \mathbf{f}_j . Therefore, we are looking for features that has a large standard deviation along with better reconstruction i.e., smaller L_{KLD} loss along with a higher standard deviation. The complete score is thus obtained as follows:

$$Score_j = \frac{L_{KLD}(\mathbf{f}_j || \tilde{\mathbf{f}}_j)}{\sigma_{\mathbf{f}_j}}; \text{ where } \sigma_{\mathbf{f}_j} \neq 0 \quad (2)$$

Once the scores are obtained, the top features are selected according to their minimum value. These top k features are then further utilized for several downstream analyses, e.g., classification and clustering. In this context, k-Nearest Neighbour (k-NN) and Random Forest (RF) classifiers are utilized for classification and the performance is quantified using Accuracy, Precision, Recall, and F1-Score. At the same time, KMeans and Agglomerative clusterings are utilized and the performance is compared using ARI, NMI, and FMI scores.

3 Results

This section describes the comparative results with respect to the state-of-the-art methods followed by a few case studies on datasets from different domains.

3.1 Comparison with the state-of-the-art method

In this subsection, the performance of the proposed model has been demonstrated over three datasets, viz., *Breast Cancer*, *Wine*, and *Digits*.

Breast Cancer:

Figure 1 describes the results related to *Breast Cancer* dataset. Figure 1A shows the embedding produced by NeuroDAVIS, which reflects those two extracted features separate the classes properly. Accordingly, the top features have been selected using NeuroDAVIS-FS, with the help of those two extracted features. After getting the efficient features, it has been compared against the state-of-the-art methods to study overlapping features (Figure 1B). 40% overlap with *RFE*; 60% overlap with *VT*, *GUS*, and *SKB*; 80% overlap with *RFECV*; and a 100% overlap with *MIC* have been observed. In order to validate the efficiency of the features selected by NeuroDAVIS-FS, classification, and clustering have been performed over the selected features. The results have again been compared against the state-of-the-art methods. It is evident from Figures 1C that both the extracted and selected features can classify the samples properly with an

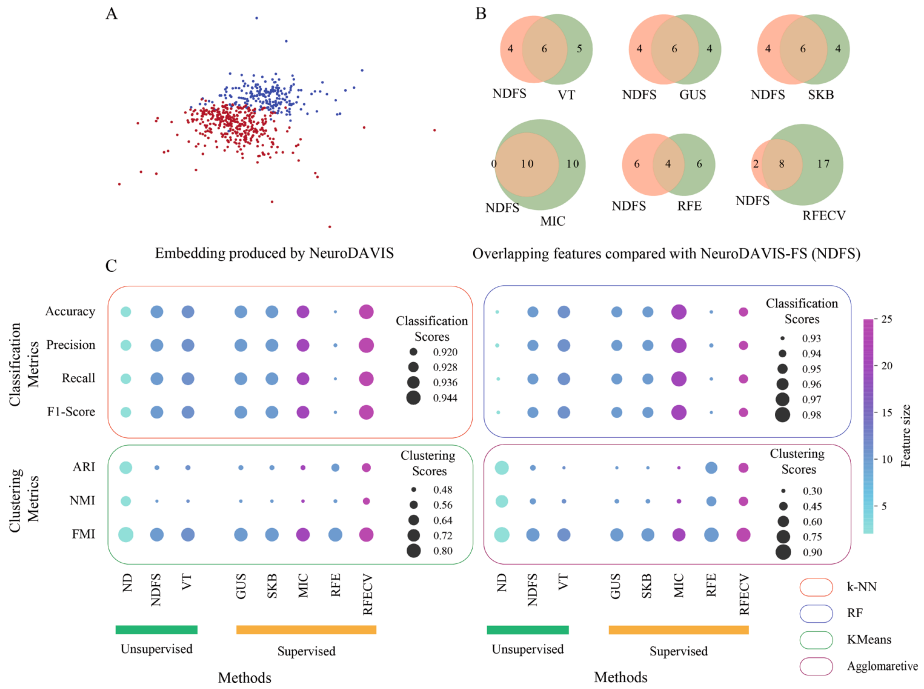


Fig. 1. For *Breast Cancer* dataset (A) The 2-dimensional embedding produced by NeuroDAVIS (ND). (B) Overlap of features between NeuroDAVIS-FS (NDFS) and the state-of-the-art. (C) Comparison of the proposed model with state-of-the-art methods based on classification and clustering performance. For classification, K-nearest neighbor (k-NN) and Random Forest (RF) models are utilized and for evaluation of clustering performance KMeans and Agglomerative clustering techniques are used.

accuracy of 0.93. *VT*, *GUS*, and *SKB* shows similar classification performance. *MIC* and *RFECV* perform better in terms of accuracy, precision, recall, and F1-score. However, the dimension of the selected feature space is significantly higher compared to the other methods. Figures 1C also shows the clustering performance of the proposed methods in terms of ARI, NMI, and FMI scores against the state-of-the-art methods. The extracted features have outperformed all the other methods and the selected features have produced comparable results in terms of clustering performance. It must be mentioned that only *VT* and the proposed approach select features in an unsupervised setup, whereas, the rest use a target class vector to do the same.

Wine:

The results on the *wine* dataset have been demonstrated using figure 2. The embedding produced by NeuroDAVIS has shown effective visualization performance in Figure 2A. Three different types of wines are well separated and clearly visible in the embedding. Using this embedding NeuroDAVIS-FS selects top fea-

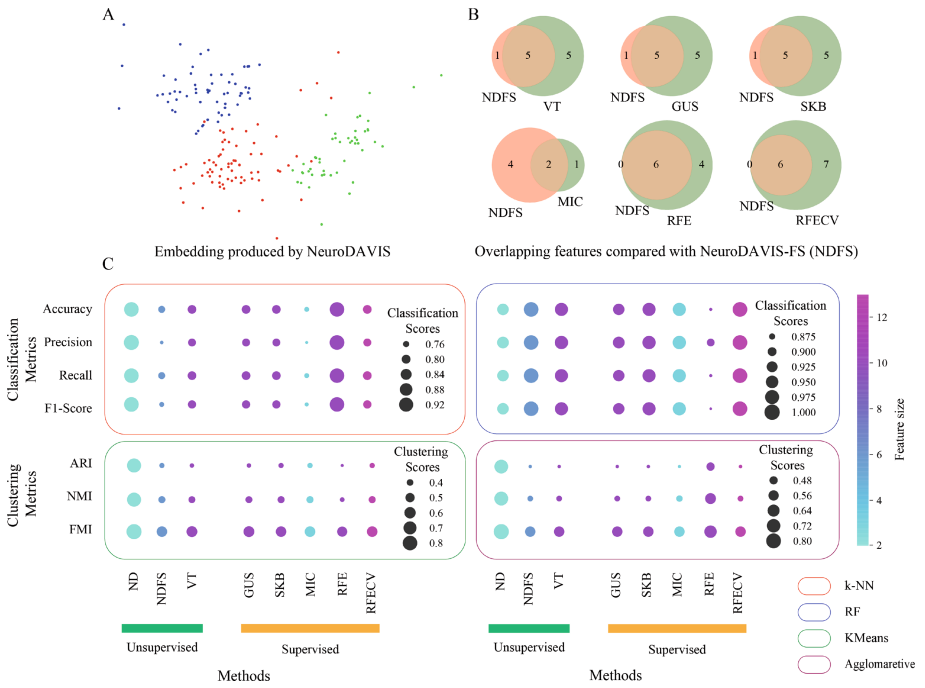


Fig. 2. For *Wine* dataset (A) The 2-dimensional embedding produced by NeuroDAVIS (ND). (B) Overlap of features between NeuroDAVIS-FS (NDFS) and the state-of-the-art methods. (C) Comparison of the proposed model with state-of-the-art based on classification and clustering performance. For classification, K-nearest neighbor (k-NN) and Random Forest (RF) models are utilized and for evaluation of clustering performance KMeans and Agglomerative clustering techniques are used.

tures. Features have also been selected using the other state-of-the-art methods and a good amount of overlapping has been found between the proposed approach and the state-of-the-art methods. 30% overlap with *MIC*; more than 80% overlap with *VT*, *GUS*, and *SKB*; and a complete overlap with *RFE* and *RFEVCV* has been observed. For *wine* dataset, ‘Alcohol’ came out as the top feature while the proposed model was used. This shows the reliability of the proposed method. Apart from the overlapping feature, classification, and clustering have been performed over *wine* dataset using two classifiers, viz., k-NN, and RF, and two clustering methods, viz., kMeans and Agglomerative clusterings respectively against the state-of-the-art approaches. It has been found from Figure 2C, that the extracted features have outperformed all the methods in terms of classification and clustering. NeuroDAVIS-FS selected top features shows a comparable result with the state-of-the-art methods by selecting comparatively less features and also by ignoring the class information.

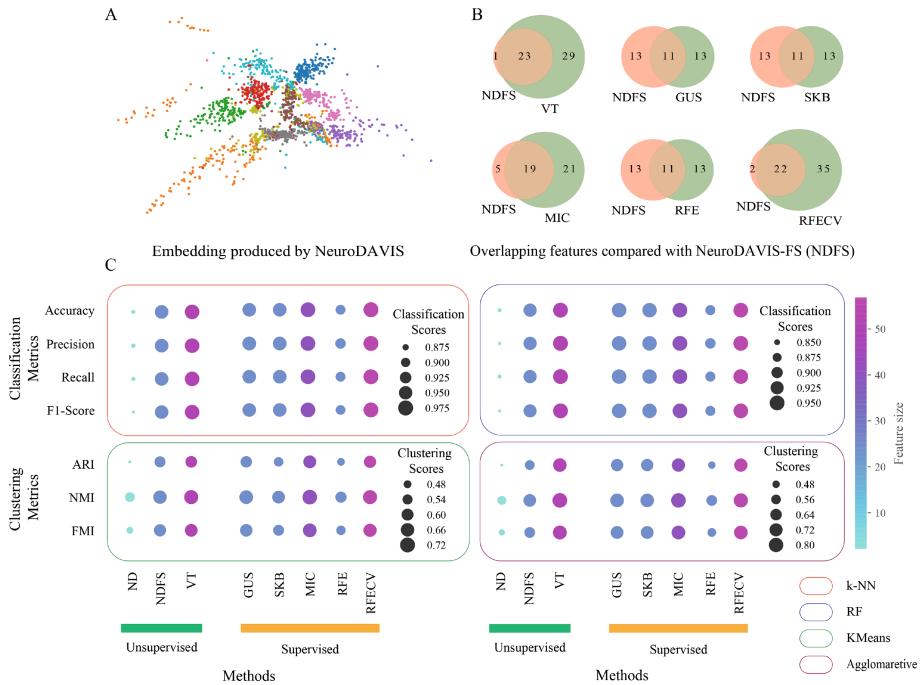


Fig. 3. For *Digits* dataset (A) The 2-dimensional embedding produced by NeuroDAVIS (ND). (B) Overlap of features between NeuroDAVIS-FS (NDFS) and the state-of-the-art methods. (C) Comparison of the proposed model with state-of-the-art based on classification and clustering performance. For classification, K-nearest neighbor (k-NN) and Random Forest (RF) models are utilized and for evaluation of clustering performance KMeans and Agglomerative clustering techniques are used.

Digits:

Finally, the effectiveness of the proposed method has been demonstrated on an image dataset viz., *Digits* dataset. NeuroDAVIS projected embedding has been shown in Figure 3A. All ten clusters are not very separated, whereas, a few compact clusters are clearly visible. Only two three clusters overlap with each other (Figure 3A). Moreover, NeuroDAVIS-FS has been applied using the embedding produced by NeuroDAVIS. It has been observed from Figure 3B, that there exists a good amount of overlap between the features selected by the proposed model and the state-of-the-art methods. 50% features are common with *GUS*, *SKB*, and *RFE*; nearly 80% features are common with *MIC*; almost a complete overlap with *VT* and *RFECV* have been observed. The proposed model has again been compared with the state-of-the-art in terms of classification and clustering. Figure 3C shows, the extracted features are not that efficient in terms of visualization, classification, and clustering. On the other hand, the selected features are performing well in terms of classification and clustering. The proposed method selects only the top 35% features and shows comparable

classification and clustering performance, however, methods like *MIC*, *VT* and *RFECV* select a lot more features to achieve the same.

3.2 Case studies

In this section, two case studies have been demonstrated on the remaining two image datasets, viz., *Coil20* and *Olivetti Faces*, and on one biological dataset. These datasets consist of a comparatively larger number of features. Moreover, the smaller sample size makes the task of feature selection even more complex.

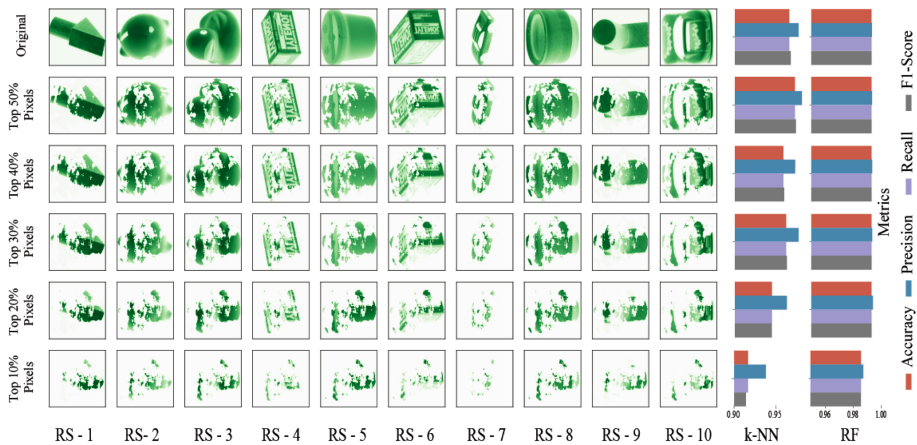


Fig. 4. Visual representation of randomly selected samples (RS) through the top features selected by the proposed model for *Coil20* dataset. The classification performance of k-NN and RF has been shown over the selected feature space.

Image datasets:

This Section discusses a case study on two image datasets. Initially, the proposed method has been examined on different sizes of top features. In other words, this analysis will help us to understand how informative the top features are. In this context, 5 sets of top features have been considered, viz., top 10%, top 20%, top 30%, top 40%, and top 50%. 10 samples have been drawn from the datasets randomly and visualized through these top features or pixels. More precisely, the randomly selected images have been visualized only through the top pixels, and the rest pixels have been set to 0. It has been observed in Figure 4 that objects are clearly identifiable through the naked eye with only 30% top features. In this regard, a classification performance has been carried out on these selected features using k-NN and RF classifiers, and it has been found that even with 10% features both the classifiers achieve an accuracy, precision, recall, and F1-score over 0.9, i.e., even those top 10% features are efficient enough to distinguish different objects almost certainly (Figure 4).

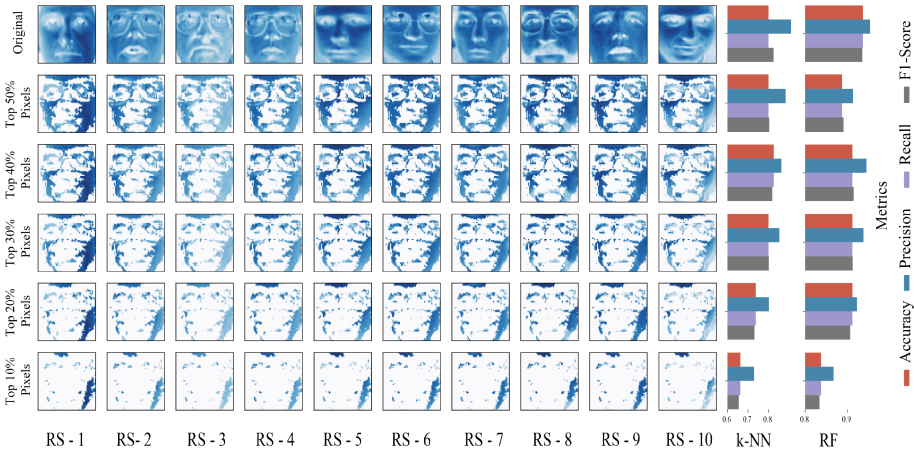


Fig. 5. Visual representation of randomly selected samples (RS) through the top features selected by the proposed model for *Olivetti Faces* dataset. The classification performance of k-NN and RF has been shown over the selected feature space.

A similar experiment has been carried out on the *Olivetti Faces* dataset. As shown in Figure 5, Unlike *Coil20* dataset, here the random samples are not identifiable through the naked eye. However, both the classifiers, k-NN and RF are able to do the same with an accuracy of at least 0.85, and 0.65 respectively. Both classifiers, with top 30% features, have shown superior classification results.

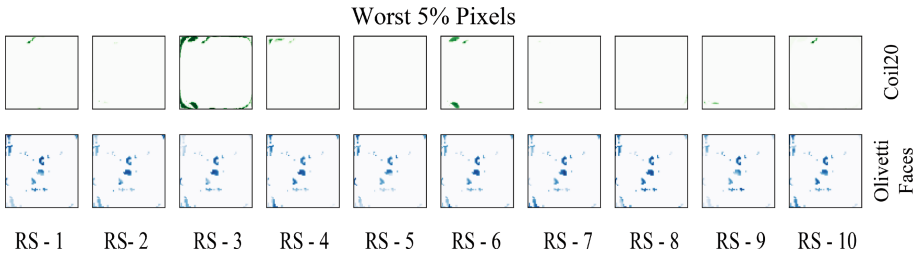


Fig. 6. Visual representation of randomly selected samples through the worst 5% features selected by the proposed model for *Coil20* (upper) and *Olivetti Faces* (lower) datasets.

In the previous experiments, it has been found that, for both the image datasets, the top features are efficient enough to perform downstream tasks with a higher accuracy. A similar experiment has been carried out on both datasets considering the top 5% worst features. It has been observed from Figure 6, that for both the datasets, the random samples look very similar, while observed through the worst features. The k-NN classifiers also reflect poor results with

an accuracy of 0.5 for *Coil20* dataset, and 0.42 for *Olivetti Faces* dataset, while applied on this worst feature space.

Biological dataset:

Finally, The performance of NeuroDAVIS-FS has been evaluated on a biological dataset, viz., *Jurkat* dataset. Initially, the dataset was pre-processed using the standard pipeline proposed in Scanpy ¹. The dataset contains gene expressions for 3388 cells across 32738 genes. Feature extraction has been performed on *Jurkat* dataset, followed by a feature selection. As shown in Figure 7 (left), the top 25 features selected by the proposed model, have been shown using a heatmap. Cluster information in the heatmap suggests that the top features can separate the major cell clusters. Moreover, a classification performance has been performed on the different sizes of top features. It has been observed from Figure 7(right), that both k-NN and RF classifiers' performance improved as the feature size decreased. k-NN classifier using only top 10% of the features produces an accuracy 0.65, and on the other hand RF classifier produces an accuracy 0.73 using top 40% features. This analysis reflects the robustness of the proposed model.

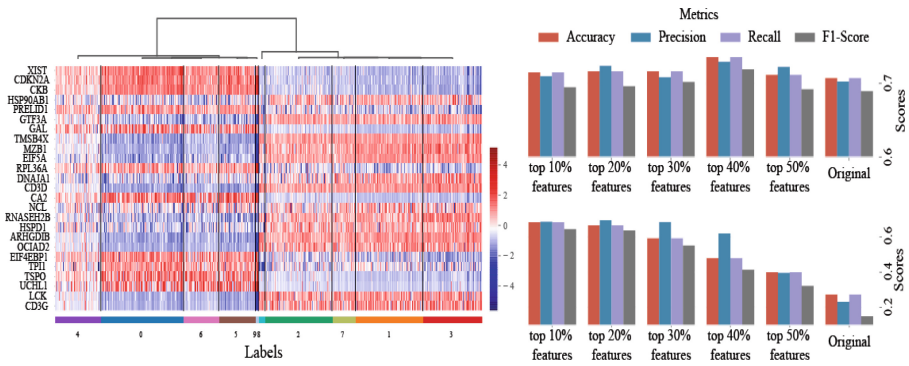


Fig. 7. Visual representation of top 25 features selected by the proposed model for *Jurkat* dataset (left). The classification performance of k-NN and RF have been shown over the selected feature spaces (right).

¹ <https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html>

4 Discussion and Conclusion

Dealing with high-dimensional datasets is challenging due to computational complexity, the curse of dimensionality, and model overfitting. It necessitates reducing the dimension of the dataset for a better understanding of inherent information and feature selection techniques are well suited to do the same. In the present study, a neural network-based model, NeuroDAVIS-FS has been proposed which is an extension of the earlier data visualization model NeuroDAVIS. The proposed feature selection method performs in an unsupervised learning setup without assuming any prior data distribution. Initially, it considers training using the model NeuroDAVIS, and selecting features according to the trained model. The efficacy of the proposed model has been demonstrated on various datasets from different domains viz., Numeric: *Breast Cancer*, and *Wine*, Image: *Digits*, *Coil20*, and *Olivetti Faces* and Biological: *Jurkat*. A comparative analysis has been executed for several downstream tasks like classification and clustering with the state-of-the-art models and it has been observed that the top features selected by the proposed model are efficient for the same. Moreover, comparable results are obtained in terms of predefined metrics for the state-of-the-art methods. Being a neural network-based approach, the model is parametric and performs in an unsupervised setup, which makes the model novel in comparison with traditional cases. In addition to that, a good amount of overlapping features have been found in the proposed feature selection model against the other classical models. Finally, two case studies on image and biological datasets with a very low sample-feature ratio (nearly 1 : 10), have been executed and found to be effective for relevant feature selection. Even with 10% features both the classifiers classify the classes of *Coil20* dataset with an accuracy of 0.9 approximately. However, for *Olivetti Faces* dataset, the accuracy rate of k-NN and RF are approximately 0.7 and 0.85 respectively. Whereas, for *Jurkat* dataset, the accuracy rate increases for the classification tasks with top 10% selected features.

Though the proposed model outperforms the state-of-the-art, it still has some limitations. As the performance of the proposed model depends on the prior training of NeuroDAVIS, the model may not be efficient in the case of biased training. Moreover, the optimal number of features has not been considered in this study, which might be an extension of the future study. Though, one can plot the feature scores in a descending order and select top features by observing the elbow. The location of a bend (knee) in the plot is generally considered as an optimal number of features.

References

1. Aeberhard, S., Coomans, D., de Vel, O.: The classification performance of rda. Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep pp. 92–01 (1992)
2. Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W., Ng, L.G., Ginhoux, F., Newell, E.W.: Dimensionality reduction for visualizing single-cell data using umap. *Nat. Biotechnol.* **37**(1), 38–44 (2019)

3. Chen, C.W., Tsai, Y.H., Chang, F.R., Lin, W.C.: Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert. Syst.* **37**(5), e12553 (2020)
4. Doquire, G., Verleysen, M.: Mutual information-based feature selection for multi-label classification. *Neurocomputing* **122**, 148–155 (2013)
5. El Touati, Y., Slimane, J.B., Saidani, T.: Adaptive method for feature selection in the machine learning context. *Engineering, Technology & Applied Science Research* **14**(3), 14295–14300 (2024)
6. Fida, M.A.F.A., Ahmad, T., Ntahobari, M.: Variance threshold as early screening to boruta feature selection for intrusion detection system. In: 2021 13th International Conference on Information & Communication Technology and System (ICTS). pp. 46–50. IEEE (2021)
7. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. vol. 4, pp. IV–317. IEEE (2007)
8. 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)
9. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* **50**(6), 1–45 (2017)
10. Liu, Y., Zheng, Y.F.: Fs_sfs: A novel feature selection method for support vector machines. *Pattern Recogn.* **39**(7), 1333–1345 (2006)
11. Maitra, C., Seal, D.B., De, R.K.: NeuroDAVIS: A neural network model for data visualization. *Neurocomputing* **573**, 127182 (2024)
12. Manikandan, G., Abirami, S.: A survey on feature selection and extraction techniques for high-dimensional microarray datasets. *Knowledge Computing and Its Applications: Knowledge Computing in Specific Domains II*, 311–333 (2018)
13. Manzoor, U., Halim, Z., et al.: Protein encoder: An autoencoder-based ensemble feature selection scheme to predict protein secondary structure. *Expert Syst. Appl.* **213**, 119081 (2023)
14. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-d objects from appearance. *Int. J. Comput. Vision* **14**(1), 5–24 (1995)
15. Mustaqim, A.Z., Adi, S., Pristyanto, Y., Astuti, Y.: The effect of recursive feature elimination with cross-validation (rfecv) feature selection algorithm toward classifier performance on credit card fraud detection. In: 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). pp. 270–275. IEEE (2021)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
17. Street, W.N., Wolberg, W.H., Mangasarian, O.L.: Nuclear feature extraction for breast tumor diagnosis. In: *Biomedical Image Processing and Biomedical visualization*. vol. 1905, pp. 861–870. SPIE (1993)

18. Venkatesh, B., Anuradha, J.: A review of feature selection and its methods. *Cybernetics and Information Technologies* **19**(1), 3–26 (2019)
19. Zheng, G.X., Terry, J.M., Belgrader, P., Ryvkin, P., Bent, Z.W., Wilson, R., Ziraldo, S.B., Wheeler, T.D., McDermott, G.P., Zhu, J., et al.: Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**(1), 1–12 (2017)



S3TC: Spiking Separated Spatial and Temporal Convolutions with Unsupervised STDP-Based Learning for Action Recognition

Mireille El-Assal[✉], Pierre Tirilly^{id}, and Ioan Marius Bilasco^{id}

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, 59000 Lille, France
{mireille.elassal2,pierre.tirilly,maris.bilasco}@univ-lille.fr

Abstract. Video analysis is a major computer vision task that has received a lot of attention in recent years. The current state-of-the-art performance in video analysis is achieved with Deep Neural Networks (DNNs) that have a high energy cost and need large amounts of labeled data for training. Spiking Neural Networks (SNNs) can have a significantly lower energy cost (thousands of times) than regular non-spiking networks when implemented on neuromorphic hardware [39,40]. They have been used for video analysis with methods like 3D Convolutional Spiking Neural Networks (CSNNs). However, these networks have a significantly larger number of parameters than spiking 2D CSNNs. This not only increases their computational cost, but can also make them more difficult to implement on ultra-low power neuromorphic hardware. In this work, we use CSNNs trained in an unsupervised manner with the Spike Timing-Dependent Plasticity (STDP) rule, and we introduce, for the first time, Spiking Separated Spatial and Temporal Convolutions (S3TCs). Using unsupervised STDP for feature learning reduces the amount of labeled data required for training. Factorizing a single spatio-temporal spiking convolution into a spatial and a temporal spiking convolution decreases the number of parameters of the network. We test our network with the KTH, Weizmann, and IXMAS datasets. Our results show that S3TCs successfully extract spatio-temporal information from videos and outperform spiking 3D convolutions, while preserving the output spiking activity, which usually decreases with deeper spiking networks.

Keywords: Spiking neural networks · STDP · Action classification · 3D convolution · Separated convolutions

1 Introduction

A large amount of new visual data is made available to the world on a daily basis, with a substantial portion of this data comprising videos. Analyzing this large amount of data is challenging for humans, which has rendered video analysis

an important computer vision task. Deep learning methods achieve state-of-the-art performance for visual data analysis. However, their computational cost is very high, which hampers their deployment on energy-constrained devices such as IoT devices; the extensive use of GPUs used to train and run them also raises some environmental concerns [29,38]. Moreover, large amounts of labeled data are needed to train them; labeling this data requires costly human intervention. This has pushed forward the exploration of methods that can analyze visual data at a lower cost. Among these methods are Spiking Neural Networks (SNNs), which are third generation neural networks capable of processing visual information in the form of low-energy spikes [40]. These networks are intended to be implemented on neuromorphic hardware [18,30,40], which is specialized hardware with the potential to overcome the limitations of traditional computing architectures, such as energy efficiency.

Neuromorphic hardware platforms like TrueNorth [31], SpiNNaker [22], Tianjic [13], BrainScaleS-2 [32], and Loihi [12] offer promising avenues, by enabling the use of SNNs at a low energy cost. In addition, analog neuromorphic hardware, based for instance on memristors or CMOS components, can offer an even lower energy consumption [23]. The latter type of neuromorphic hardware pairs well with Spike Timing-Dependent Plasticity (STDP) learning methods [36,37] which are used to train SNNs in an unsupervised manner. This learning method is still immature and does not achieve the classification rates of other methods like ANN-to-SNN conversion [9] and surrogate gradient-based methods [11], but it has the potential to mitigate some of their limitations, such as the need for substantial labeled data and the use of global computations, which are difficult to implement on ultra-low power neuromorphic hardware. STDP can be used to learn features from the data in an unsupervised fashion prior to classification, reducing the need for labeled data; in addition, as a purely local learning rule, it limits the communication overhead within neuromorphic circuits, making them easier to design.

Some spiking models have been proposed for video analysis, including spiking two-stream methods [44], spiking ResNets [19], 2D Convolutional Spiking Neural Networks (CSNNs), and 3D CSNNs [15]. While most spiking models, similarly to 2D CSNNs, require non-spiking pre-processing [14] for motion extraction, 3D CSNNs [15] have the advantage of being fully-spiking solutions to motion pattern learning. However, similarly to traditional methods, spiking 3D convolutions increase the number of trainable parameters with respect to spiking 2D convolutions. This can make their implementation on neuromorphic hardware more challenging, as the larger number of parameters results in a larger number of connections to be fit in the circuit design. Integrating separated convolutions [41] into STDP-based SNN training could mitigate the issues of 3D convolutions: this architecture conserves the ability to directly learn spatio-temporal patterns, but with fewer connections, thanks to the splitting of convolutional filters into a series of filters of lower dimensions. It makes it a promising approach for implementing SNNs on neuromorphic hardware efficiently. Separated convolu-

tions have already shown promising results with CNNs [8,27,33] but have not yet been used in the spiking domain.

In this work, we present Spiking Separated Spatial and Temporal Convolutions (S3TCs), where we factorize a spiking spatio-temporal 3D convolution into two separate smaller spatial and temporal convolutions. We use CSNNs trained with the unsupervised STDP learning rule. S3TCs are expected to be more efficient and hardware friendlier solutions. To the best of our knowledge, our work is the first to address the subject of separated convolutions trained with STDP. We hypothesize that the benefits of separated convolutions with CNNs could apply to SNNs, and that lower-dimensional filters could improve STDP learning by capturing more generic patterns and prompting the neurons to fire more spikes. This work is a building block towards improving the performance of spiking models that can learn spatio-temporal features from video data. The main contributions of this paper are summarized as follows:

- we present Spiking Separated Spatial and Temporal Convolutions (S3TCs);
- we evaluate the performance of S3TC models with different filter sizes on the KTH [34], Weizmann [24], and IXMAS [43] datasets;
- we compare the performance of S3TCs to that of spiking 3D convolution from [15], and we conclude that S3TCs can achieve better performance;
- we show that factorizing the 3D filters into two sets of 2D and 1D filters, with STDP, may lead to learning more generic patterns;
- we show that S3TCs provide a deeper spiking network than 3D CSNNs while maintaining a similar output spiking activity, which is critical to enable the design of deeper spiking architectures.

2 Related Work

3D CNNs are a common practice for motion modeling [1,3,4,20,21,25]. The third dimension of these networks, which is devoted to time, enables the extraction of spatio-temporal features. In [42], the authors present deep 3D CNNs for spatio-temporal feature learning. They compare them to 2D CNNs, and conclude that 3D architectures perform better for video analysis. However, these models have more trainable parameters than 2D models, which consequently increases their computational cost and makes the optimization of these parameters more difficult. To mitigate this problem, one solution is separable convolutions. With separable convolutions, a convolution filter is separated into two or more filters of smaller dimensionalities, each filter typically processing a distinct subset of dimensions of the initial filter. Separable convolutions adopted in networks like MobileNets [27] and Xception [10] have succeeded in decreasing the number of parameters of these networks while preserving their performance. Moreover, gains in accuracy have been recorded when factorizing a 3D convolution into a 2D spatial convolution and a 1D temporal convolution [41]. In [41], the authors attribute this gain in accuracy to additional nonlinearities added by the separated convolutions compared to using a 3D convolution. They argue that these nonlinearities render the model capable of representing more complex functions.

They also add that 2D and 1D filters are easier to optimize than 3D filters, where appearance and dynamics are intertwined. However, the aforementioned work do not address the case of spiking neural networks.

CSNNs provide a cost-effective and unsupervised alternative for motion modeling. 3D CSNNs have been proposed recently [15]. In [15], the authors use unsupervised 3D CSNNs trained with STDP to learn spatio-temporal visual features for action classification; they conclude that 3D CSNNs outperform 2D CSNNs at learning visual features used in human action recognition, especially with longer video sequences. However, despite the energy efficiency of these 3D CSNN models compared with traditional non-spiking methods, the additional parameters, with regard to 2D CSNNs, result in additional operations and potentially more complex neuromorphic hardware [12]. To the best of our knowledge, spiking separable convolutions have not yet been explored in the literature; however, they could be an option to mitigate the issues of spiking 3D convolutions and enable the design of video analysis models that can be implemented more easily on low-power neuromorphic hardware.

3 Spiking Convolutions and Network Architecture

We build on the standard recognition pipeline introduced in [18], which is effective in object classification, and has been applied successfully to action recognition by including suited pre-processing [14] or 3D spiking convolutional layers [15]. This choice enables an accurate comparison of the performance between our S3TCs and the 3D convolution model presented in [15]. The classification pipeline (see Figure 1) includes visual feature learning by an SNN trained with STDP, which helps reduce the reliance on annotated data, and a final classifier that performs classification based on the extracted features. In this paper, we focus on efficiently training the SNN with STDP to learn spatio-temporal visual features. In the following, we describe the major components of this pipeline.

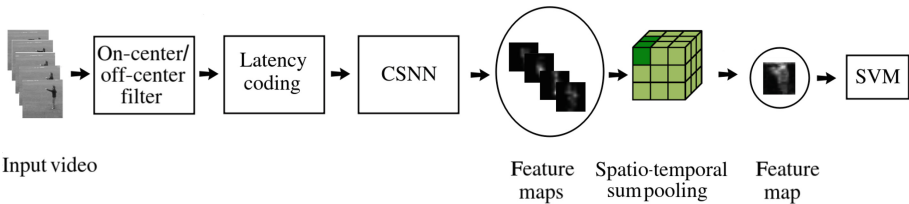


Fig. 1. Pipeline for action recognition with unsupervised feature learning by a CSNN

3.1 Input Video Samples

A video is a sequence of frames represented as a 4D tensor of size $l_w \times l_h \times l_c \times l_{td}$ where l_w and l_h are the width and height of the frames, l_c is their number of

channels, which is 1 in the case of grayscale frames, and l_{td} is the temporal depth of the tensor i.e., the number of frames in the video sample.

3.2 Neuron Model and Training

The SNNs used in this work consist of Integrate-and-Fire (IF) neurons [5]. The IF neuron model is characterized by its internal state $v(t)$, called membrane potential. Spikes incoming from the input synapses are integrated by the neuron and increase its membrane potential, as follows:

$$f_s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$v(t) = \sum_{n \in \mathcal{N}} W_n \cdot f_s(t - t_n) \quad (2)$$

where f_s is the kernel of spikes, \mathcal{N} is the set of spikes incoming from the input synapses, W_n is the weight of the synapse transmitting spike n , t_n is the timestamp at which spike n reaches the neuron, and t is the current time. f_s defines spikes as impulses localized in time, which increase the potential of the neuron when they are integrated. The IF neuron is also characterized by its threshold $v_{th}(t)$. When the membrane potential $v(t)$ of the neuron reaches or exceeds $v_{th}(t)$, the neuron generates an output spike, and its membrane potential is reset to its resting potential v_r , which we set to 0 in this work.

Training is unsupervised and uses the biological STDP learning rule [35]. Given a synapse at the input of a neuron, biological STDP updates the weight of the synapse when the neuron fires an output spike, as follows:

$$\Delta W = \begin{cases} +\eta_w \cdot e^{-\frac{t_{\text{post}} - t_{\text{pre}}}{\tau}} & \text{if } t_{\text{pre}} \leq t_{\text{post}} \\ -\eta_w \cdot e^{-\frac{t_{\text{pre}} - t_{\text{post}}}{\tau}} & \text{if } t_{\text{pre}} > t_{\text{post}} \end{cases} \quad (3)$$

where t_{pre} is the timestamp of the input spike incoming from the synapse, t_{post} is the timestamp of the output spike fired by the neuron, τ is a time constant that controls the magnitude of the update in time, η_w is the learning rate, and ΔW is the update applied to the neuron, so that $W := W + \Delta W$. STDP increases the weights of synapses from which input spikes came right before the output spike was fired (they are considered as the cause of the output spike), and decrease the weights of synapses from which input spikes came right after (they are considered as unrelated to the output spike). Over updates, STDP makes the synapses converge towards a specific pattern of correlated input spikes. As STDP updates synapses independently, it is a local training rule and is not affected by changes in neuron connectivity.

A layer trained with STDP uses Winner-Takes-All (WTA) inhibition during training to prevent several neurons from learning the same pattern: the first neuron to fire an output spikes prevents the other neurons in the layer from spiking until the end of the sample presentation. With WTA, some neurons can

overpower other neurons, i.e., they have a tendency to fire more spikes than others. This leads to the layer being stuck in a state where a few active neurons fire all the time, while the others are quiet. To balance and control training within the layer, we use the threshold adaptation method introduced in [18]. This method updates neuron thresholds $v_{\text{th}}(t)$ as follows:

$$v_{\text{th}}(t) := v_{\text{th}}(t) + \Delta v_{\text{th}} + \Delta v'_{\text{th}} \quad (4)$$

$$\Delta v_{\text{th}} = \begin{cases} \eta_{\text{th}} & \text{if the neuron is the first to fire within the layer} \\ \frac{\eta_{\text{th}}}{n_l} & \text{otherwise} \end{cases} \quad (5)$$

$$\Delta v'_{\text{th}} = -\eta_{\text{th}} \cdot (t - \hat{t}) \quad (6)$$

where η_{th} is the threshold learning rate, n_l is the number of neurons in layer l , t is the current timestamp, and \hat{t} is a manually defined target timestamp. Term Δv_{th} increases the threshold of the neuron that just fired, so that they are become less likely to fire, and decreases the thresholds of others to promote firing; it balances training. Term $\Delta v'_{\text{th}}$ adjusts the thresholds so that neurons tend to fire at a specific timestamp \hat{t} ; it gives better control over the patterns to be learned and increases the quality of the visual features [18].

3.3 3D Spiking Convolution

A 3D convolutional layer has f_k trainable filters, with sizes $f_w \times f_h \times f_{\text{td}}$, where f_w and f_h represent the width and height of the filter respectively, and f_{td} is the temporal size of the filter. During the convolution operation, these filters slide along the temporal dimension of a video sample, in addition to the spatial ones.

Each neuron of a layer is connected to $f_w \times f_h \times f_{\text{td}}$ neurons of the previous layer. The membrane potential of 3D spiking convolutional neurons can be expressed as shown in Equation 7 from [15]:

$$v_{x,y,z,k}(t) = \sum_{n \in \mathcal{N}} W_{i(x_n),j(y_n),m(z_n),k_n,k} \times f_s(t - t_n) \quad (7)$$

where f_s is the kernel of spikes (see Eq. 1), $v(t)$ is the potential of the neuron membrane at time t , and x , y , z , and k are the coordinates of the spike in the width, height, time, and channel dimensions, respectively. \mathcal{N} is the set of input connections in the neighborhood, $W \sim U(0, 1)$ is the trainable synaptic weight matrix, $i()$, $j()$, and $m()$ are functions that are used to map the location of the input neuron to the corresponding location in the weight matrix, and k_n is the index of the trainable filter. When the membrane potential $v_{x,y,z,k}(t)$ crosses the threshold potential $v_{\text{th}}(t)$, the synaptic weights and thresholds of the network are updated according to Equations 3, 4, 5, and 6.

The number of parameters in one 3D spiking convolutional layer is:

$$|P| = f_k \times n_c \times f_w \times f_h \times f_{\text{td}} \quad (8)$$

where P represents the set of parameters in the model, f_k is the number of filters, n_c is the number of input channels, f_w and f_h represent the width and height of the filter, respectively, and f_{td} is the temporal size of the filter.

3.4 Spiking Separated Spatial & Temporal Convolutions

With separated convolutions, the filter connectivity of the spiking 3D convolution layer introduced in Section 3.3 can be broken down into two parts, space-wise and time-wise convolutions, as shown in Figure 2.

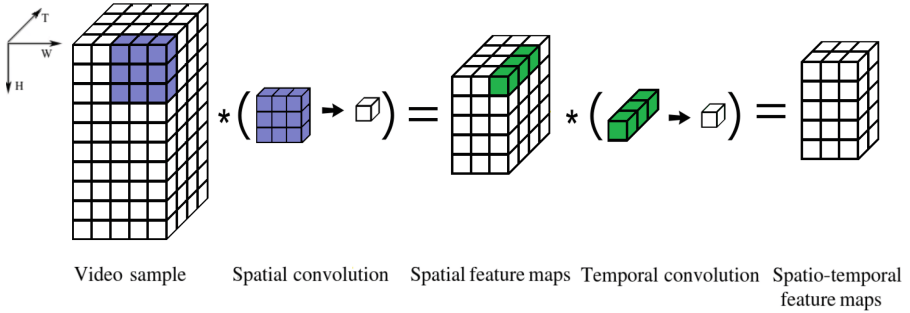


Fig. 2. Separable spatial and temporal convolutions

In the first phase, a 2D filter slides over the spatial dimension of the input, one frame at a time. This filter has a dimension of $f_w \times f_h \times 1$, and results in spatial feature maps. In the second phase, with time-wise convolution, we compute a linear combination of the spatial feature maps by undergoing a $1 \times 1 \times f_{td}$ convolution in the temporal dimension to extract meaningful temporal information from the spatial feature maps. S3TC can be formalized as Equation 9 for the space-wise convolution, and Equation 10 for the time-wise convolution:

$$v_{x,y,k}^S(t) = \sum_{n \in \mathcal{N}^S} W_{i(x_n),j(y_n),k_n,k} \times f_s(t - t_n) \tag{9}$$

$$v_{z,k}^T(t) = \sum_{n \in \mathcal{N}^T} W_{m(z_n),k_n,k} \times f_s(t - t_n) \tag{10}$$

where $v^S(t)$ and $v^T(t)$ are the membrane potential at time t of the neurons of the spatial and the temporal convolutions, respectively, and \mathcal{N}^S and \mathcal{N}^T are the sets of input connections in the spatial and temporal neighborhoods, respectively.

The number of parameters in S3TC layers is:

$$|P'| = f_k \times n_c \times (f_w \times f_h + f_{td}) \tag{11}$$

This number of parameters is lower than that of a spiking 3D convolution, which reduces the number of operations required to train them, and can reduce the connection overhead in low-power hardware implementations. In the next section, we study the trade-off between accuracy and efficiency with these two spiking convolution settings.

3.5 Baseline Classification Pipeline

The baseline pipeline is shown in Figure 1. The video frames are filtered with an on-center/off-center filter [2], which uses a Difference-of-Gaussian (DoG) filter used to pre-process the data by simulating on-center/off-center cells and extracting edges. This filter is needed because STDP-based SNNs need edges to learn informative patterns [16]. After that, latency coding is applied to transform the edges into spikes represented by timestamps. Larger edge values are represented as earlier spikes, while lower values come later. Next comes the CSNN processing, which can be a 3D CSNN as mentioned in Section 3.3 or a S3TC network. The output of our network will consist of spatio-temporal feature maps, which are then reduced in size using spatio-temporal sum-pooling before being sent to a classifier. We use a Support Vector Machine (SVM) for classification as it yields good performance with default parameters, but any other classifier could be used. We do not use a spiking classifier because we focus on unsupervised feature learning with STDP, and single-spike supervised classification with SNN is still emerging, so such a classifier could make the results harder to interpret.

4 Evaluation

This section contains the details of our experiments. First, we present the datasets, along with the implementation details and the main parameters of our network. Then we present the results of implementing and testing our S3TCs, and we compare them to spiking 3D convolutions.

4.1 Datasets and Evaluation Protocol

We use three datasets: the KTH [34], Weizmann [24], and IXMAS [43] datasets. The KTH and Weizmann datasets are early and simple datasets for action recognition. Although traditional computer vision approaches have already achieved high recognition rates on these datasets [6], their simplicity makes them good basic benchmarks to evaluate emerging models like the ones targeted in this paper. The IXMAS dataset features different actors, cameras, and viewpoints, which adds complexity. Moreover, its settings are more challenging, as two thirds of the recordings contain objects in the scene, partially occluding the actors.

The KTH dataset contains 600 videos that feature 25 subjects performing 6 actions in 4 scenarios. Subjects 11, 12, 13, 14, 15, 16, 17, and 18 are used for training, while 19, 20, 21, 23, 24, 25, 01, 04 are used for validation, and 02, 03, 05, 06, 07, 08, 09, 10, and 22 are used for testing, as indicated in the KTH protocol.

The Weizmann dataset contains 90 videos of 9 subjects performing 10 actions. The experiments on this dataset all use the leave-one-subject-out (LOSO) strategy. In this approach, models are trained on data from 8 subjects and tested on the remaining subject. This process is repeated for each subject, ensuring that the model is evaluated on completely unseen individuals, which better simulates real-world scenarios, where the system needs to generalize to new users.

Similarly, the IXMAS action recognition dataset consists of 10 subjects, 11 actions, and 1148 sequences. The experiments on this dataset also follow the LOSO strategy, where each subject is left out in turn for testing, while the model is trained on the remaining subjects.

To shorten the running time of experiments, we take subsets of the video frames, like in [1], [28], and [15]. We use 10 frames per video, and skip three frames between each two selected frames in order to make sure to capture a full cycle of the performed action. We also scale down the frame sizes to half of their original sizes to increase the processing speed.

We measure the classification accuracy (in %) on the test set for all experiments. Each experiment was run ten times, and we report the mean and standard deviation of accuracy over the ten runs.

4.2 Implementation Details

The video is pre-processed with the on-center/off-center filter mentioned in Section 3.5. This filter has a size of 7×7 , and uses centered isotropic Gaussians of variance 1.0 and 4.0.

The 3D CSNN consists of a single layer, whereas the S3TC is composed of one 2D layer followed by one 1D layer. Convolutional layers have $f_k = 64$ filters for both 3D and S3TC settings.

Neuron thresholds are randomly initialized with a normal distribution, which has a mean of 8 and variance of 0.1 for all experiments except those with a filter size of 3, where we decrease the mean to 5. This is because small filters integrate fewer input spikes, resulting in no spiking activity when the threshold is too high. The value of the target timestamp \hat{t} discussed in Section 3.2 are taken from [15]: we use a value of $\hat{t} = 0.65$ for the KTH and IXMAS datasets, and a value of $\hat{t} = 0.75$ for the Weizmann dataset.

Spatio-temporal pooling is set to limit the size of the output feature maps to $20 \times 20 \times 2$. Then, the output feature maps are linearized and introduced into an SVM with a linear kernel, which performs action classification. The default hyperparameters of libSVM [7] are used.

The software simulator used to simulate the convolutional SNNs tested in this work is the [CSNN simulator](#) [16], which is a publicly available and open-source simulator. The source code for our experiments will be released publicly as a specific branch of the CSNN simulator.

4.3 3D vs. Separable Convolutions

We test 3D convolutions and S3TCs for five different filter sizes: $f \in [3, 5, 7, 9, 10]$. For the sake of limiting the possible filter size combinations, we use the same size $f = f_h = f_w = f_{td}$ for all dimensions. A 3D convolution has filters of size $f \times f \times f$, while the filter sizes of its corresponding separated convolutions are

$f \times f \times 1$ for the spatial convolution and $1 \times 1 \times f$ for the temporal one. $f = 3$ is the most common kernel size in the literature [10, 27, 33]; however, larger filter sizes like 5 and 7 have shown to give better results in [26], so we elected to perform experiments over a range of filter sizes.

Table 1. Classification rates in % (average \pm standard deviation) for the KTH, Weizmann, and IXMAS datasets (10 frames per video) over 10 runs with 3D convolution and separated convolutions. Bold indicates the best performance for each dataset.

(A) Filter size = 3		
Dataset	3D Conv	Separated Conv
KTH	65.79 \pm 0.0073	67.69 \pm 0.0021
Weizmann	61.56 \pm 0.0107	62.20 \pm 0.0100
IXMAS	53.96 \pm 0.0083	52.41 \pm 0.0027
(B) Filter size = 5		
Dataset	3D Conv	Separated Conv
KTH	67.59 \pm 0.0041	69.21 \pm 0.0037
Weizmann	63.20 \pm 0.0096	66.83 \pm 0.0217
IXMAS	50.88 \pm 0.0034	52.22 \pm 0.0032
(C) Filter size = 7		
Dataset	3D Conv	Separated Conv
KTH	68.52 \pm 0.0000	71.48 \pm 0.0021
Weizmann	62.85 \pm 0.0095	63.94 \pm 0.0115
IXMAS	39.90 \pm 0.0024	48.68 \pm 0.0029
(D) Filter size = 9		
Dataset	3D Conv	Separated Conv
KTH	67.59 \pm 0.0000	64.84 \pm 0.0000
Weizmann	64.62 \pm 0.0189	65.79 \pm 0.0187
IXMAS	34.25 \pm 0.0042	44.63 \pm 0.0031
(E) Filter size = 10		
Dataset	3D Conv	Separated Conv
KTH	59.86 \pm 0.0050	62.22 \pm 0.0132
Weizmann	62.16 \pm 0.0310	59.38 \pm 0.0209
IXMAS	28.22 \pm 0.0040	38.59 \pm 0.0035

Table 1 presents the accuracy of the classification pipeline for each data set and convolution filter size, for 3D convolution and S3TC. These results show that S3TC achieves better performance than 3D convolution in 12 out of 15 configurations, while having less parameters, thus requiring less operations. With large

enough filter sizes (e.g., 5 and 7), S3TC can outperform regular 3D convolution for most datasets. This behavior is similar to the results in [41], where the authors indicate that 2D and 1D filters are easier to optimize with supervised learning than 3D filters. In our case, we use unsupervised STDP, so it cannot be the reason for the improvement in performance. As STDP tends to converge quickly, 3D convolutions may learn complex sample-specific patterns, while separated convolutions would learn more generic patterns, involving less parameters, that get combined sequentially. This in part improves the learning, in addition to the increased depth of the S3TC network, which introduces more non-linearity.

Additionally, the results demonstrate that different datasets have different optimal filter sizes (7 for the KTH dataset, 3 for IXMAS, and 5 for Weizmann). This variation is related to the nature of the datasets: the KTH dataset has a higher resolution (160×120) than IXMAS (48×64), and it features subjects at a larger scale than the Weizmann dataset, so it needs a larger filter size to learn optimal features. Scale also matters over the temporal dimension: larger temporal filters provide a better extraction of moving patterns with datasets that exhibit significant variations or movements, like the KTH dataset, while smaller filters are needed for datasets that exhibit smaller variations, like the Weizmann dataset. Therefore, the performance of S3TCs, similarly to 3D spiking convolutions, depends greatly on the size of the convolutional filters.

To further explain these results, we measured the activity of the CSNNs at their outputs. Figure 3 shows the number of output spikes generated by each CSNNs for filter sizes $f \in [3, 5, 7, 9]$ on the KTH dataset. Results show that the output activity is similar for 3D convolutions and S3TCs. Spiking activity in SNNs typically decreases as more layers are added [17]. However, in our case, the two layers of S3TCs maintain similar activity levels compared to a single layer of 3D convolutions. It indicates that each layer of S3TCs tends to respond to more patterns of the input than a single layer of 3D convolutions, which confirms that it learns more generic patterns. Since the loss of activity is a major issue in designing deeper SNNs, the fact that S3TCs have the same impact on spiking activity than 3D convolutions, despite their additional layer, means that using them should not prevent from using deeper architectures. Figure 3 also shows that the spiking activity at the output decreases uniformly as filter sizes increase. So, the optimal filter size is not dependent on the specific spiking activity of the network or the sparsity of the resulting feature vectors; it confirms that the choice of the best filter size depends mostly on the spatial and temporal properties of actions featured in the dataset.

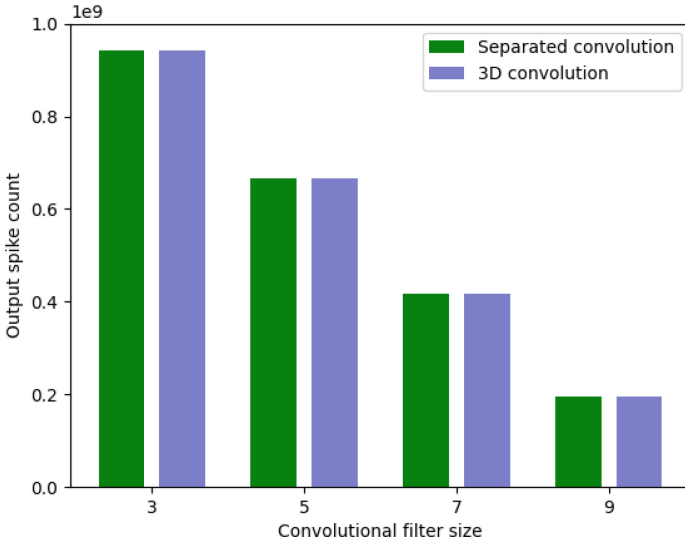


Fig. 3. Final output number of spikes fired by separable spatial and temporal convolutions compared to 3D convolutions with filter sizes of 3, 5, 7 and 9 using the KTH dataset.

5 Conclusion

Convolutional spiking neural networks can offer an energy-efficient solution to computer vision tasks on neuromorphic hardware. Especially, 3D CSNNs have been shown to be effective for action classification. However, using 3D convolutions, which are suitable for video analysis, increases the number of parameters, making training more challenging and potentially leading to more complex hardware requirements. To mitigate this issue, we chose to reduce the number of parameters in the network by replacing spiking 3D convolutions with spiking separated convolutions: we factorize a single 3D spiking convolution into two separate spatial and temporal spiking convolutions. This separation decreases the number of parameters, and can improve the performance when using sufficiently large filters. The difference in performance between 3D convolutions and separable convolutions is highly dependent on choosing suited filter size.

One conclusion is that the optimal filter size varies from one dataset to another depending on the scale of motion in space and time. A second conclusion is that S3TCs can outperform 3D convolutions thanks to their network being deeper, which adds more non-linearity, and to the lower dimension of their filters, which allows STDP to converge towards more generic patterns. Although S3TCs are deeper spiking networks than 3D CSNNs, their output spiking activity does not decrease, making them suited to the design of deeper architectures.

A promising avenue for future work would involve using a multi-stream architecture with S3TC networks, each stream using a specific filter size. This approach would bring invariance to scale, enabling the capture of information about

both small and large motion patterns, leading to better generalization across different datasets.

Acknowledgments. This work has been supported by IRCICA (USR 3380) under the bio-inspired project, and funded by Région Hauts-de-France. Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

References

1. Arunnehr, J., Chamundeeswari, G., Bharathi, S.P.: Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos. *Procedia Computer Science* **133**, 471–477 (2018). <https://doi.org/10.1016/j.procs.2018.07.059>
2. Babaiee, Z., Hasani, R.M., Lechner, M., Rus, D., Grosu, R.: On-Off Center-Surround Receptive Fields for Accurate and Robust Image Classification. In: *International Conference on Machine Learning (ICML)*. pp. 1–21 (2021)
3. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential Deep Learning for Human Action Recognition. In: *International Workshop on Human Behavior Understanding (HBU)*. pp. 29–39 (2011). https://doi.org/10.1007/978-3-642-25446-8_4
4. Belmonte, R., Ihaddadene, N., Tirilly, P., Bilasco, I.M., Djeraba, C.: Video-Based Face Alignment With Local Motion Modeling. In: *Winter Conference on Applications of Computer Vision (WACV)*. pp. 2106–2115 (2019). <https://doi.org/10.1109/WACV.2019.00228>
5. Burkitt, A.: A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input. *Biological Cybernetics* **95**, 1–19 (2006). <https://doi.org/10.1007/s00422-006-0068-6>
6. Chakraborty, B., Holte, M., Moeslund, T., González, J., Roca, X.: A Selective Spatio-temporal Interest Point Detector for Human Action Recognition in Complex Scenes. In: *International Conference on Computer Vision (ICCV)*. pp. 1776–1783 (2011). <https://doi.org/10.1109/ICCV.2011.6126443>
7. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1–27:27 (2011). <https://doi.org/10.1145/1961189.1961199>, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
8. Chen, J., Lu, Z., Liao, Q.: XSepConv: Extremely Separated Convolution for Efficient Deep Networks with Large Kernels. In: *International Conference on Digital Image Processing (ICDIP)*. vol. 11878 (2021). <https://doi.org/10.1117/12.2601043>
9. Chen, Q., Rueckauer, B., Li, L., Delbruck, T., Liu, S.C.: Reducing Latency in a Converted Spiking Video Segmentation Network. In: *IEEE International Symposium on Circuits and Systems (ISCAS)* (2021). <https://doi.org/10.1109/ISCAS51556.2021.9401667>
10. Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 1251–1258 (2017)

11. Dampfhofer, M., Mesquida, T., Valentian, A., Anghel, L.: Backpropagation-Based Learning Techniques for Deep Spiking Neural Networks: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–16 (2023). <https://doi.org/10.1109/TNNLS.2023.3263008>
12. Davies, M., Srinivasa, N., Lin, T.H., Chinya, G., Cao, Y., Choday, S.H., Dimou, G., Joshi, P., Imam, N., Jain, S., Liao, Y., Lin, C.K., Lines, A., Liu, R., Mathaikutty, D., McCoy, S., Paul, A., Tse, J., Venkataramanan, G., Weng, Y.H., Wild, A., Yang, Y., Wang, H.: Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* **38**(1), 82–99 (2018). <https://doi.org/10.1109/MM.2018.112130359>
13. Deng, L., Wang, G., Li, G., Li, S., Liang, L., Zhu, M., Wu, Y., Yang, Z., Zou, Z., Pei, J., Wu, Z., Hu, X., Ding, Y., He, W., Xie, Y., Shi, L.: Tianjic: A Unified and Scalable Chip Bridging Spike-Based and Continuous Neural Computation. *IEEE J. Solid-State Circuits* **55**(8), 2228–2246 (2020). <https://doi.org/10.1109/JSSC.2020.2970709>
14. El-Assal, M., Tirilly, P., Bilasco, I.M.: A Study On the Effects of Pre-processing On Spatio-temporal Action Recognition Using Spiking Neural Networks Trained with STDP. In: *International Workshop on Content-based Multimedia Indexing (CBMI)* (2021). <https://doi.org/10.1109/CBMI50038.2021.9461922>
15. El-Assal, M., Tirilly, P., Bilasco, I.M.: 2D versus 3D Convolutional Spiking Neural Networks Trained with Unsupervised STDP for Human Action Recognition. In: *International Joint Conference on Neural Networks (IJCNN)* (2022). <https://doi.org/10.1109/IJCNN55064.2022.9892063>
16. Falez, P.: Improving Spiking Neural Networks Trained with Spike Timing Dependent Plasticity for Image Recognition. Ph.D. Thesis, Université de Lille (2019), <https://hal.archives-ouvertes.fr/tel-02429539>
17. Falez, P., Tirilly, P., Bilasco, I.M., Devienne, P., Boulet, P.: Mastering the Output Frequency in Spiking Neural Networks. In: *International Joint Conference on Neural Networks (IJCNN)* (2018). <https://doi.org/10.1109/IJCNN.2018.8489410>
18. Falez, P., Tirilly, P., Marius Bilasco, I., Devienne, P., Boulet, P.: Multi-layered Spiking Neural Network with Target Timestamp Threshold Adaptation and STDP. In: *International Joint Conference on Neural Networks (IJCNN)* (2019). <https://doi.org/10.1109/IJCNN.2019.8852346>
19. Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., Tian, Y.: Deep Residual Learning in Spiking Neural Networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. vol. 34, pp. 21056–21069 (2021)
20. Feichtenhofer, C.: X3D: Expanding Architectures for Efficient Video Recognition. In: *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
21. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast Networks for Video Recognition. In: *International Conference on Computer Vision (ICCV)* (2019). <https://doi.org/10.1109/ICCV.2019.00630>
22. Furber, S.B., Galluppi, F., Temple, S., Plana, L.A.: The SpiNNaker Project. *Proc. IEEE* **102**(5), 652–665 (2014). <https://doi.org/10.1109/JPROC.2014.2304638>
23. Gao, B., Zhou, Y., Zhang, Q., Zhang, S., Yao, P., Xi, Y., Liu, Q., Zhao, M., Zhang, W., Liu, Z., Li, X., Tang, J., Qian, H., Wu, H.: Memristor-based analogue computing for brain-inspired sound localization with in situ training. *Nature Communications* **13**, 2026 (04 2022). <https://doi.org/10.1038/s41467-022-29712-8>
24. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(12), 2247–2253 (2007). <https://doi.org/10.1109/ICCV.2005.28>

25. Hara, K., Kataoka, H., Satoh, Y.: Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? CoRR **abs/1711.09577** (2017), <http://arxiv.org/abs/1711.09577>
26. Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., Le, Q.: Searching for MobileNetV3. In: International Conference on Computer Vision (ICCV). pp. 1314–1324 (2019). <https://doi.org/10.1109/ICCV.2019.00140>
27. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR **abs/1704.04861** (2017), <http://arxiv.org/abs/1704.04861>
28. Ji, S., Xu, W., Yang, M., Yu, K.: 3D Convolutional Neural Networks for Human Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013). <https://doi.org/10.1109/TPAMI.2012.59>
29. Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the Carbon Emissions of Machine Learning. CoRR **abs/1910.09700** (2019), <http://arxiv.org/abs/1910.09700>
30. Lee, C., Panda, P., Srinivasan, G., Roy, K.: Training Deep Spiking Convolutional Neural Networks With STDP-Based Unsupervised Pre-training Followed by Supervised Fine-Tuning. *Frontiers in Neuroscience* **12** (2018). <https://doi.org/10.3389/fnins.2018.00435>
31. Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S.K., Appuswamy, R., Taba, B., Amir, A., Flickner, M.D., Risk, W.P., Manohar, R., Modha, D.S.: A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface. *Science* **345**(6197), 668–673 (2014). <https://doi.org/10.1126/science.1254642>
32. Pehle, C., Billaudelle, S., Cramer, B., Kaiser, J., Schreiber, K., Stradmann, Y., Weis, J., Leibfried, A., Müller, E., Schemmel, J.: The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity. *Frontiers in Neuroscience* **16** (2022). <https://doi.org/10.3389/fnins.2022.795876>
33. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted residuals and linear bottlenecks. In: International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4510–4520 (2018). <https://doi.org/10.1109/CVPR.2018.00474>
34. Schuld, C., Laptev, I., Caputo, B.: Recognizing Human Actions: A Local SVM Approach. In: International Conference on Pattern Recognition (ICPR). p. 32–36 (2004). <https://doi.org/10.1109/ICPR.2004.1334462>
35. Schuman, C.D., Potok, T.E., Patton, R.M., Birdwell, J.D., Dean, M.E., Rose, G.S., Plank, J.S.: A Survey of Neuromorphic Computing and Neural Networks in Hardware. CoRR **abs/1705.06963** (2017), <http://arxiv.org/abs/1705.06963>
36. Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., Linares-Barranco, B.: STDP and STDP Variations with Memristors for Spiking Neuromorphic Learning Systems. *Frontiers in Neuroscience* **7** (2013). <https://doi.org/10.3389/fnins.2013.00002>
37. Singha, A., Muralidharan, B., Rajendran, B.: Analog Memristive Time Dependent Learning Using Discrete Nanoscale RRAM Devices. In: International Joint Conference on Neural Networks (IJCNN). pp. 2248–2255 (2014). <https://doi.org/10.1109/IJCNN.2014.6889915>

38. Strubell, E., Ganesh, A., McCallum, A.: Energy and Policy Considerations for Deep Learning in NLP. In: Annual Meeting of the Association for Computational Linguistics (ACL). pp. 3645–3650 (2019). <https://doi.org/10.18653/v1/P19-1355>
39. Sun, Y., Zeng, Y., Li, Y.: Solving the Spike Feature Information Vanishing Problem in Spiking Deep Q Network With Potential Based Normalization. *Frontiers in Neuroscience* **16** (2022). <https://doi.org/10.3389/fnins.2022.953368>
40. Tavanaei, A., Ghodrati, M., Kheradpisheh, S.R., Masquelier, T., Maida, A.: Deep Learning in Spiking Neural Networks. *Neural Netw.* **111**, 47–63 (2019). <https://doi.org/10.1016/j.neunet.2018.12.002>
41. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6450–6459 (2018). <https://doi.org/10.1109/CVPR.2018.00675>
42. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning Spatiotemporal Features With 3D Convolutional Networks. In: International Conference on Computer Vision (ICCV) (2015). <https://doi.org/10.1109/ICCV.2015.510>
43. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint Action Recognition Using Motion History Volumes. *Comput. Vis. Image Underst.* **104**(2–3), 249–257 (2006). <https://doi.org/10.1016/j.cviu.2006.07.013>
44. Zhang, J., Wang, J., Di, X., Pu, S.: High-Accuracy and Energy-Efficient Action Recognition with Deep Spiking Neural Network. In: International Conference on Neural Information Processing (ICONIP). pp. 279–292 (2022). https://doi.org/10.1007/978-3-031-30108-7_24



Recommendation of Data-Free Class-Incremental Learning Algorithms by Simulating Future Data

Eva Feillet^{1,2} , Adrian Popescu¹ , and Céline Hudelot² 

¹ Université Paris-Saclay, CEA, LIST, 91120 Palaiseau, France
adrian.popescu@cea.fr

² Université Paris-Saclay, CentraleSupélec, MICS, Gif-sur-Yvette, France
{eva.feillet, celine.hudelot}@centralesupelec.fr
<https://list.cea.fr/en/>, <https://www.mics.centralesupelec.fr/>

Abstract. Class-incremental learning deals with data streams composed of batches of classes. Various algorithms have been proposed to address the challenging case where samples from past classes cannot be stored. However, selecting an appropriate algorithm for a user-defined setting is an open problem, as the relative performance of these algorithms depends on the incremental setting. To solve this problem, we introduce an algorithm recommendation method that simulates the future data stream. Given an initial set of classes, our method leverages generative models to simulate future classes from the same visual domain. We evaluate recent algorithms on the simulated stream and recommend the one that performs best in the user-defined incremental setting. We illustrate the effectiveness of our method on three large datasets using six algorithms and six incremental settings. Our method performs close to an oracle that would choose the best algorithm in each setting. This work contributes to facilitating the practical deployment of continual learning.

Keywords: Continual learning · Recommendation · Image classification

1 Introduction

Continual learning (CL) aims at building models able to handle new data or tasks over time [5, 32, 33]. Class-incremental learning (CIL), in which the data stream is composed of batches of classes [44], is an actively studied CL paradigm [2, 25, 46]. At each step of a CIL process, the model is updated with a new batch of classes while attempting to maintain the performance on all previously learned classes. In many practical applications of CL, computational costs and memory budgets are important constraints [10, 13]. *The data-free version of CIL (DFCIL)*

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_21.

has gained attention because it requires lower storage [16,52], making it suitable for resource-constrained applications on embedded devices. It is also a relevant paradigm when training data cannot be stored for privacy reasons [40]. Recent comparative studies [7,30] have shown that none of the DFCIL approaches proposed to date is the best for all practical cases. The performance of DFCIL algorithms depends on the characteristics of the incremental process, e.g., the number of incremental steps, the number of classes per step, and the amount of training data available per class. *Given this variability in performance, we aim to recommend an appropriate algorithm for a user-specified DFCIL scenario.* This recommendation problem was addressed in [7] using a large set of precomputed experiments run on benchmark datasets. While interesting, this method requires many precomputed experiments and does not account for the visual domain of the classification task.

In this article, we tackle the recommendation of DFCIL algorithms from a data-centric point of view. Our method, illustrated in Figure 1, takes as inputs the settings of the DFCIL process (number of incremental steps, number of classes per step) and a subset of classes available at the start of the process. Given a set of DFCIL algorithms, the recommended algorithm is obtained by:

1. building a data stream that simulates future classes belonging to the same visual domain as the initial classes,
2. evaluating the candidate DFCIL algorithms on the simulated data stream,
3. recommending the algorithm that performs best on the simulated stream.

Our recommendation method is evaluated on three large datasets using six competitive DFCIL algorithms in six incremental scenarios. The results show that the performance of our method is close to that of an oracle that selects the best algorithm in any scenario. By simulating the future stream of data either using generative models or using the visual knowledge base ImageNet21k, our recommendation method compares favorably to the fixed choice of any DFCIL algorithm we tested, and AdvisIL [7]. Additionally, we propose a strategy to lower the cost of exploring the performance of all candidate algorithms on the simulated data stream. Code is available at <https://github.com/EvaJF/SimuGen>.

2 Related work

2.1 CIL algorithms

The ability to continually learn from new data is needed to develop more autonomous and sustainable AI systems [5,33]. One of the main challenges of CIL is to mitigate *catastrophic forgetting* [2,8], namely the tendency of CIL models to abruptly forget previously acquired information when confronted with new information. To cope with this issue, numerous approaches have been proposed [2,25], based on parameter-isolation [1,36], iterative fine-tuning with distillation [19,24,51], classifier-incremental learning with a fixed representation [9,12,29] and more recently, dynamic prompting of transformer models [21,47].

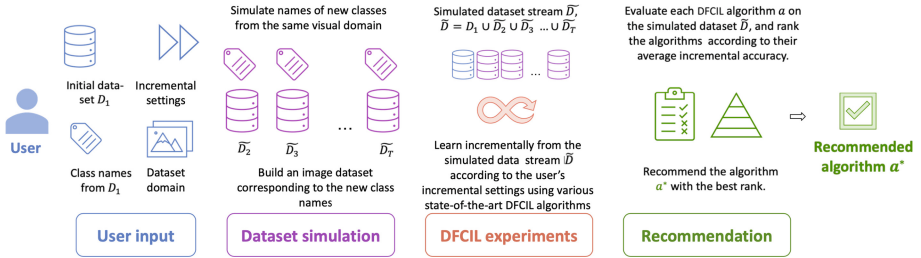


Fig. 1. Method overview. A user needs an algorithm for a given DFCIL use case. He has access to an initial labeled dataset D_1 and provides expected characteristics of the incremental process, e.g. the number of classes per step. Based on these inputs, our method simulates a data stream \bar{D} extending D_1 , first by proposing future class names, then by populating these new classes with images. Next, it evaluates various DFCIL algorithms on the simulated stream and recommends the one with the best performance for deployment with real data.

Catastrophic forgetting is particularly challenging in DFCIL, as optimizing the parameters of a deep neural network to recognize new classes without examples of past classes skews the classifier towards new classes [19]. Some DFCIL methods combine fine-tuning with knowledge distillation [17] to alleviate forgetting [20, 50, 51]. They handle new classes well since model parameters are updated to fit the novelty, but despite distillation, they tend to have lower performance for past classes [25]. Another line of work proposes to use a fixed feature extractor trained during the initial step and focuses on incrementally learning only a classifier [9, 12, 29]. In this case, the challenge lies in leveraging the fixed representation to separate all past and new classes well. In our experiments, we include both types of methods to assess their strengths and limitations.

2.2 Generative models

We explore the potential of using generative models to simulate a future data stream. We use this stream to simulate a class-incremental learning process. Note that our goal is not to develop a competitor to existing generative models, but to use them in a novel way for DFCIL.

In natural language processing, state-of-the-art large language models (LLMs) are based on transformer architectures [3] that are trained in a self-supervised manner. LLMs such as T5 [31], Llama-v2 [42] or the family of GPT models [3] show impressive results in generative tasks such as summarizing a text or writing a story. In this work, we use an LLM to generate class names related to the initial classes of the user. In computer vision, multimodal models can generate images from textual prompts. Diffusion Probabilistic Models [39] (DPMs), based on a series of cascading denoising auto-encoders, currently achieve state-of-the-art results. The popular Stable Diffusion (SD) method [34] improves the scalability of DPMs by performing denoising from a lower-dimensional space

than the pixel space. SD models are designed with a general-purpose conditioning mechanism and can be prompted with textual descriptions.

Previous works [37, 41] use diffusion models to create synthetic datasets for transfer learning purposes. Image generation was used in continual learning but for generative replay [20, 38]. Recently, the authors of [20] proposed to generate images of past classes by prompting SD at each step of the CIL process with the names of past classes. This method achieves competitive performance but assumes that a large diffusion model can be used throughout the CIL process, an assumption that contradicts the frugality requirements (memory, computation, latency) generally associated with CIL applications [2, 5]. In this article, we take inspiration from [37] to simulate a dataset in the context of DFCIL. Unlike [20], *we only use generative models in the initial non-incremental step of the CIL process*. This approach is compatible with CIL because data generation is carried out offline, for simulation purposes only, and is not part of the incremental process.

2.3 Simulating data streams for CL

In [4], an algorithm is introduced to rearrange samples from a given dataset to form a data stream whose distribution changes continuously rather than in discrete steps as in batch-wise CIL. In [15], a sampling-based generator creates arbitrarily long data streams with control over the repetition of past classes via probability distributions. These two approaches [4, 15] focus on evaluating streaming algorithms, whereas many DFCIL algorithms handle data that arrive in batches without repetition.

The work that is closest to ours is AdvisIL [7]. This method uses a set of precomputed DFCIL experiments done with auxiliary datasets to simulate incremental processes. The main limitation of AdvisIL is that the user-defined DFCIL settings must be similar to those of a subset of the precomputed experiments. Furthermore, AdvisIL’s recommendations do not consider the semantic characteristics of the incremental datasets, such as their visual domain or the granularity of the classification tasks.

3 DFCIL process

In this section, we remind the DFCIL paradigm. We consider a dataset $\mathcal{D} = D_1 \cup D_2 \cup \dots \cup D_T$ and a sequential learning process composed of T non-overlapping steps s_1, s_2, \dots, s_T . A step s_i consists of learning from the labeled samples of the set of new classes P_i from the subset D_i . Each sample from D_i belongs to a unique class from P_i , and each class is present in a single data subset. We denote by n the average number of images per class in D_1 . We consider that each of P_2, P_3, \dots, P_T has the same number of classes N .

Model training. At the first step s_1 , the model \mathcal{M}_1 is trained on the data subset D_1 that involves the set of classes P_1 . For each of the following steps, the same procedure is applied. For $i = 2, 3, \dots, T$, at the step s_i , the model \mathcal{M}_i first

recovers the parameters from the model \mathcal{M}_{i-1} obtained in the previous step. It is then updated using the samples from D_i to incorporate the new classes of P_i . Depending on the DFCIL algorithm, only some of the model parameters are updated (e.g. only those of the classifier in [9, 12, 29], those of the classifier and some of the feature extractors in [28], or all parameters in [19, 24]). We experiment with algorithms covering these three strategies in Section 5.

Evaluation. CIL algorithms should be designed for arbitrarily long incremental processes and any number of classes per update. In practice, for evaluation purposes, T and N are defined based on the datasets used in the evaluation benchmarks [2, 25, 44]. DFCIL algorithms are commonly evaluated based on their *average incremental accuracy*, computed as $A = \frac{1}{T} \sum_{i=1}^T q_i$, where q_i is the accuracy of the model \mathcal{M}_i on the test samples from $\bigcup_{j=1}^i D_j$, after learning from the training samples of D_i at step s_i .

4 Method

We introduce a method that recommends a DFCIL algorithm according to the characteristics of the incremental process and an initial dataset D_1 . We explain our working hypotheses in Subsection 4.1. The first step of our method is to build a simulated dataset that will be used as a proxy for the future data stream. We present two approaches for building such a simulated dataset in Subsection 4.2. Finally, we present in Subsection 4.3 how to recommend a DFCIL algorithm by evaluating candidate algorithms on the simulated dataset.

4.1 Working hypotheses

The first hypothesis made in this work relates to the characteristics of the incremental process. While our method can be applied in any setting, it also inherits the evaluation-related constraints of the algorithm it compares [2, 25, 44]. As the update frequency of models must be decided in advance for evaluation, we assume that the user provides an estimate of the number of incremental steps T and an estimate of the number of classes per step N . We note that our method can also be a way to experiment with different data distribution scenarios.

Second, we make the usual supervised learning assumption that the class labels from P_1 are known. In practice, we associate each class name with a description as we observed in preliminary experiments that more descriptive prompts improve the conditioning of image generation with SD, confirming the results of [37]. Class descriptions are either generated automatically by an LLM or retrieved from a knowledge base (more details in Appendices A and B).

Third, following existing CIL works [2, 11, 12], *we distinguish between the first non-incremental step s_1 and the following steps and consider that compute and memory constraints apply after the initial step.* Under this assumption, we use generative models to simulate the future stream and to recommend a suitable algorithm before deployment in a user-specified scenario. In addition, in the illustration of our recommendation method in Section 6, *to ensure a fair comparison*

of algorithms, we experiment with DFCIL algorithms with comparable memory requirements at inference.

Finally, following common CIL benchmarks using ImageNet subsets [6, 9] or fine-grained thematic datasets as in [25], we assume that classes belong to a common topic or domain. This is a reasonable assumption in many applications, e.g. LandSnap for landmark identification. We note that our data simulation method is not bound to this hypothesis, i.e. the generative models can be prompted to introduce a domain shift or a different topic if the user wants to, but we make this choice to circumscribe our evaluation.

4.2 Building simulated datasets

We denote a simulated dataset by $\tilde{\mathcal{D}} = \tilde{\mathcal{D}}_1 \cup \tilde{\mathcal{D}}_2 \cup \dots \cup \tilde{\mathcal{D}}_T$, where $\tilde{\mathcal{D}}_1$ is the initial user dataset ($\tilde{\mathcal{D}}_1 = \mathcal{D}_1$) and for $i \geq 2$, each subset $\tilde{\mathcal{D}}_i$ corresponds to N new classes with n images per class. As specified in Section 3, the subsets of classes of $\tilde{\mathcal{D}}$ do not intersect. In the following, we describe two approaches to obtaining $\tilde{\mathcal{D}}$, which will be a potential future stream with classes from the same visual domain as \mathcal{D}_1 . To control the semantic content of the simulated stream, each approach begins by constructing a set of $(T - 1) \cdot N$ new class names $\tilde{P}_2 \cup \tilde{P}_3 \cup \dots \cup \tilde{P}_T$, ensuring that these new classes do not appear in the initial set of classes P_1 . Then, each class is populated with either generated or real images.

We carried out preliminary experiments simulating future classes using geometric data augmentation, such as rotations or mixing of class pairs. However, the simulated data streams were too far from the actual data distribution to obtain relevant recommendations (see Appendix C.3 for details).

Generative simulation. Our first approach, named **SimuGen**, uses an LLM and a text-to-image model. The LLM can provide visual descriptions for the class names from P_1 when not readily available in a resource such as WordNet [26]. We first build a list L of pairs of the form (c, d) , where c is the name of a class in P_1 and d is its associated description. In the prompt, the description d facilitates disambiguation. We also observe that asking for a visual description of the items produces more relevant suggestions for class names. Then, building on the representation of the visual task provided by the initial class names and their descriptions, we aim to obtain $(T - 1) \cdot N$ classes within the same visual domain as \mathcal{D}_1 . An LLM can produce many different class names, and the prompts can be advantageously tweaked to obtain class names more or less similar to those in P_1 . The textual output of the LLM allows us to form a new list of pairs of the form (c', d') , with c' a new class name and d' its associated description.

In practice, we choose LLaMav2-13b-chat [42] as it balances performance and inference time. From a sublist of 3 class names from P_1 , their description and the visual domain of the user, we prompt the LLM with the following pattern: “Here is a list of [visual domain]: [sublist]. Could you provide ten more items on the same topic, with a short visual description of each item?”. Since the input and output lengths of the LLM are limited, we prompt it multiple times with different sublists instead of once with the entire list, and we ask for ten new items at each time. Diversifying the sublists also diversifies the results.

To facilitate postprocessing, we use a system prompt asking for a JSON output. We iterate the process until we obtain $N \cdot (T - 1)$ new unique class names to extend the initial subset of class names P_1 .

The second step of SimuGen consists of generating n images for each new class name using a text-to-image model. For a given class, we obtain its associated images by prompting the model with the class name c' and its description d' obtained in the first step of SimuGen.

In practice, we use Stable-Diffusion-2-1-base that provides high-quality images. We prompt the model with the following pattern using n different random seeds to obtain n images for each new class: "a [style] photo of a [class name], [description]", where **style** is selected from a domain-related list. A prompt example used to generate an image is "a panorama photo of Salar de Uyuni, the world's largest salt flat, Bolivia". As reported in [37], associating class names with some context produces better image diversity and avoids the pitfalls of rare or ambiguous words. We provide more details about the use of LLaMAv2-13b-chat and Stable Diffusion in Appendices B and C, respectively.

Simulation using a knowledge base. Our second approach, named **Proxy21k**, selects new classes from an existing large-scale and general-purpose visual dataset. In our experiments, we use the ImageNet dataset [6], which covers the concepts from WordNet lexical database [26]. ImageNet has a hierarchical structure and includes over 21,000 classes. We prune ImageNet to keep only subtrees of the visual database related to the domain of the initial dataset. Then, we randomly pick $N \cdot (T - 1)$ new classes among the ImageNet leaf classes from the preselected subtrees having at least n images. One underlying assumption of Proxy21k is that ImageNet-21k allows the sampling of a sufficiently big simulated dataset that covers the same visual domain as \mathcal{D} . This assumption is strong when ImageNet-21k does not sufficiently cover the target visual task.

4.3 Recommending a DFCIL algorithm

Let \mathcal{A} be a set of candidate DFCIL algorithms. We consider that a simulated dataset $\tilde{\mathcal{D}}$ was created using either the SimuGen or Proxy21k approach. We consider three recommendation strategies that use $\tilde{\mathcal{D}}$.

Greedy recommendation. This strategy consists in evaluating the performance of each algorithm in \mathcal{A} on the T steps of the simulated stream and recommending the algorithm with the best average incremental accuracy on $\tilde{\mathcal{D}}$.

Efficient recommendations. We propose two alternative recommendation strategies to reduce the computational cost of simulations. We consider the case where only $t < T$ steps can be simulated. (i) We propose to run all candidate algorithms during the first t simulation steps and to recommend the algorithm that achieves the best average accuracy at the end of these t steps.

(ii) We also propose to first run t simulation steps and, after each simulation step $k \geq t$, until a single algorithm remains or T is reached, to discard from the set of candidate algorithms \mathcal{A}_k the algorithm denoted $a_{(k)}^-$ whose current average accuracy over the simulated steps is the lowest, i.e. $a_{(k)}^- = \operatorname{argmin}_{a \in \mathcal{A}_k} \sum_{i=1}^k q_i$,

where q_i is the accuracy on the test samples from $\bigcup_{j=1}^i D_j$ of the model \mathcal{M}_i trained using algorithm a , after learning at step s_i . In our experiments, we set $t = 3$.

We remind that, like AdvisIL [7], our method recommends a DFCIL algorithm adapted to user-provided incremental learning settings. Unlike AdvisIL, we adopt a data-centric point of view to personalize the recommendation: (i) we take into account the semantic content of the user’s dataset, and (ii) our recommendation method is not based on a set of precomputed experiments, so it can provide relevant recommendations whatever the incremental settings.

5 Evaluation framework

Reference datasets. We experiment with the following reference datasets : ILSVRC [35], iNaturalist 2018 [43], and Google Landmarks v2 [27]. We sample from them three balanced 1000-class subsets denoted IN1k, iNat1k, and Land1k, with 350, 310, and 330 images per class, respectively (see Appendix A for more details). These datasets cover diversified visual tasks and allow us to assess the advantages and limitations of the proposed simulation approaches.

DFCIL algorithms. We consider a set \mathcal{A} composed of six candidate DFCIL algorithms. Four of them rely on a fixed feature extractor and learn new classifiers incrementally. NCM [32] uses a nearest-class mean classifier, DSLDA [12] a streaming LDA, FeTrIL [29] linear SVCs, and FeCAM [9] a Bayesian classifier based on the Mahalanobis distance. PlaStIL [28] freezes a part of the backbone and combines fine-tuning of the last layers with linear SVCs. BSIL [19] is a fine-tuning-based algorithm that relies on a weighted softmax loss to rebalance predictions between old and new classes. Note that we performed preliminary experiments (on Land1k) with other fine-tuning-based methods, namely SDC [50], and PASS [51], but they were not recommended in any of the tested settings, confirming previous results reported in [9, 29]. After also considering the high computational cost of training SDC and PASS we did not retain them for the main experiments since their inclusion would not change the findings.

To ensure comparability, all algorithms are implemented with the ResNet18 architecture [14] commonly used in CIL [25, 32]. Given a dataset and an initial number of classes, all algorithms share the same initial model (trained with the BSIL code). Note that for a fair comparison regarding the amount of data stored, we use the version of FeCAM that stores a single covariance matrix for all classes. Please refer to Appendix D for further implementation details.

DFCIL scenarios. The DFCIL algorithms are evaluated in six incremental scenarios that push them to their limits. Three scenarios follow the protocol of [32] in the case of a 1000-class dataset: 50 steps of 20 classes each, 10 steps of 100 classes each, or 5 steps of 200 classes each. The other scenarios follow the protocol of [18], where the initial dataset contains half of all classes. The remaining classes are split into 5, 10, or 100 steps (of resp. 100, 50, or 5 classes).

Performance. We run each algorithm in each incremental scenario on each real dataset (IN1k, iNat1k, and Land1k) and their corresponding simulated

datasets obtained with SimuGen and Proxy21k. We denote by “oracle” the method that always selects the best-performing algorithm on the real dataset. We aim for the recommendation to behave like the oracle. In Table 1, Δ_{simu}^{strat} indicates the difference between the average incremental accuracy of the algorithm provided by the oracle and that of the recommended algorithm when using the stream simulation *simu* (either SimuGen (*gen*) or Proxy21k (*proxy*)) in combination with the recommendation strategy *strat*, which is denoted (i) "T" for the greedy recommendation strategy, (ii) "3" for the strategy simulating only the first 3 incremental steps or (iii) "3+" for the strategy simulating 3 incremental steps and then discarding the least performing algorithm until only one remains, or *T* is reached. Finally, we denote by Δ_m the accuracy gap between the oracle and a fixed baseline, which always recommends an algorithm *m*.

Table 1. Performance gap (Δ) between the average incremental accuracy of the methods proposed by the oracle (A_{ref}) and that of the algorithms recommended with different methods. Results averaged over (i) the three datasets (ii) the six DFCIL settings of the form $(Card(P_1), T)$, (iii) all settings. Individual results are in Appendix E. P: PlaStIL, B: BSIL, N:NCM, D:DSLDA, F:FeTrIL, Fc: FeCAM. *Gaps closer to zero are better.* Best results in bold, second best underlined.

		Accuracy gap: recommendation methods vs oracle													
		Oracle	SimuGen + reco.			Proxy21k + reco.			Baselines: AdvsiL and fixed reco.						
		$A_{ref}(\%)$	Δ_{gen}^T	Δ_{gen}^{3+}	Δ_{gen}^3	Δ_{proxy}^T	Δ_{proxy}^{3+}	Δ_{proxy}^3	Δ_{Adv}	Δ_P	Δ_B	Δ_N	Δ_D	Δ_F	Δ_{Fc}
Scenario	(20, 49)	35.12	0.0	-3.95	-3.95	0.0	-2.55	-6.50	0.0	-11.68	-11.54	-17.00	<u>-2.31</u>	-4.35	0.0
	(100, 10)	57.22	<u>-1.41</u>	0.0	0.0	-2.93	-1.66	-1.66	-4.11	-6.33	0.0	-12.88	-6.89	-5.26	-4.11
	(200, 5)	66.94	0.0	0.0	0.0	<u>-1.10</u>	<u>-1.10</u>	<u>-1.10</u>	-7.30	-5.73	0.0	-11.44	-10.54	-7.12	-7.30
	(500, 5)	71.06	<u>-0.14</u>	0.0	0.0	-2.64	-1.99	-1.86	-2.84	-6.05	0.0	-3.14	-6.18	-3.66	-2.84
	(500, 11)	67.98	-0.26	-0.26	-3.50	-0.25	<u>-0.22</u>	-1.05	-0.05	-9.73	-3.45	<u>-0.22</u>	-3.21	-1.65	-0.05
	(500, 101)	67.8	-0.11	<u>-0.16</u>	-0.89	-0.27	<u>-0.16</u>	<u>-0.16</u>	-0.11	-60.25	-59.78	<u>-0.16</u>	-3.12	-3.63	-0.11
Dataset	IN1k	49.07	-0.08	-1.98	-3.18	<u>-0.71</u>	-0.08	-1.98	-1.18	-14.20	-11.25	-5.09	-3.22	-3.00	-1.18
	INat1k	60.82	<u>-0.07</u>	-0.02	-0.44	-0.39	-1.27	-1.71	-3.01	-18.45	-12.21	-7.08	-6.58	-4.48	-3.01
	Land1k	73.17	-0.81	-0.19	<u>-0.55</u>	-2.49	-2.49	-2.47	-3.02	-17.23	-13.93	-10.26	-6.32	-5.35	-3.02
Average		61.02	-0.32	<u>-0.73</u>	-1.39	-1.20	-1.28	-2.06	-2.40	-16.63	-12.46	-7.47	-5.37	-4.28	-2.4

6 Results

Main results. The results presented in Table 1 show that the recommendations based on SimuGen and Proxy21k are effective, since they recommend algorithms whose accuracy on the real stream \mathcal{D} is close to or equal to that of the best algorithm (oracle). In most settings, they also perform better than the considered baselines. On average, the best scores are obtained with SimuGen, whose recommendations are only 0.32 accuracy points below the oracle when all simulation steps are run. Proxy21k also gives interesting results, but the gap with the oracle is higher than for SimuGen (1.20pts). A closer look at the results shows that

the data streams simulated with SimuGen are closer to the real data than with Proxy21k, especially in the case of Land1k. This is illustrated by the example in Figure 2a, where the SimuGen simulations better fit the experiments with the reference datasets than the Proxy21k simulations. The same is true for iNat1k (Figure 2b) despite this dataset being well covered by ImageNet-21k, the visual knowledge base used by Proxy21k. We conclude that *simulation using generative models enables a better and more flexible approximation of the incremental data stream than using a preexisting database*. **Baselines.** Our incremental settings are closer to the same subset of AdvisIL experiments for which FeCAM performs better, so here AdvisIL always recommends FeCAM. The precomputed experiments presented in AdvisIL [7] focus on incremental settings with at most 100 classes and smaller neural architectures, advantaging fixed-representation algorithms over fine-tuning-based algorithms. This highlights AdvisIL’s lack of flexibility beyond its initial configurations.

Among the baselines that recommend a fixed algorithm (see Table 1, baselines), FeCAM has the best accuracy on average, followed by FeTrIL and DSLDA. These methods perform better when the feature extractor is trained on a larger subset of classes because the resulting deep representation is more transferable. We note that FeCAM separates classes particularly well, as it can handle the heterogeneity of scales for the different class distributions. Figure 2 shows that its initial accuracy is close to that of FeTrIL and BSIL while its accuracy is more stable. DSLDA also has stable performance across the incremental process. On the contrary, BSIL performs well on the initial classes as its representation is optimized for this very task. BSIL performs better when the number of incremental steps is small. Despite a knowledge distillation loss and a rebalanced softmax loss to emulate past classes, this algorithm is penalized by the difficulty of preserving an adapted representation of past classes when the incremental sequence is long. While NCM is the simplest method, it still outperforms BSIL and PlaStIL in the challenging scenario ($Card(P_1) = 500, T = 101$), where it benefits from a highly transferable feature extractor. Although on par with BSIL and DSLDA on short scenarios, PlaStIL has the lowest average accuracy across all experiments. Its partial fine-tuning mechanism without knowledge distillation is pushed to its limits by long sequences of tasks.

Scenarios. On average, the most challenging scenario for all methods is the one with 50 steps with 20 new classes per step (Figure 2b). The representation learned from the 20 initial classes hardly generalizes to the rest of the stream, and fine-tuning struggles with the numerous small incremental steps.

Recommendation dynamics. In Figure 3, for an incremental setting of the form $(Card(P_1), T)$, we display the accuracy gap between the best algorithm (oracle) and the recommendations made after performing only $t \leq T$ simulation steps. The relevance of recommendations evolves with the number of incremental learning steps performed on the simulated streams. Except in the case $(Card(P_1) = 100, T = 10)$ where BSIL is the best algorithm for all three datasets from the initial step, Figure 3 shows the interest of recommending an algorithm using either Proxy21k or SimuGen, even if the DFCIL experiments

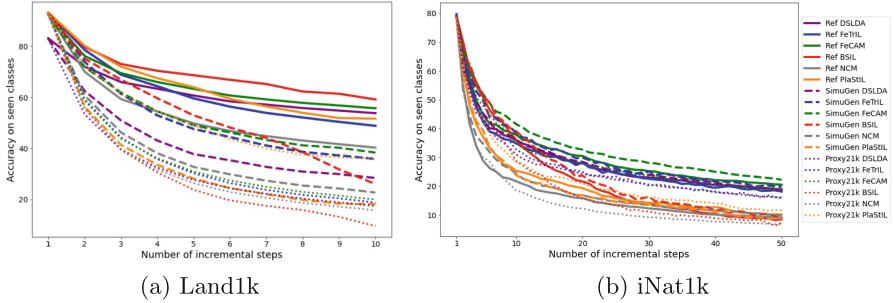


Fig. 2. Detailed incremental accuracy for (a) iNat1k with $\text{Card}(P_1) = 20$ and $T = 50$ steps and (b) Land1k with $\text{Card}(P_1) = 100$ and $T = 10$ steps, and their corresponding simulated datasets obtained following SimuGen and Proxy21k.

are only partially executed. In Table 1, we see that the efficient recommendation strategies consisting in either running only three steps (Δ_{simu}^3) or running three steps then pruning the set of candidate algorithms (Δ_{simu}^{3+}), also perform better than the choice of any fixed algorithm.

We provide more detailed plots and tables in Appendix E, showing the relevance of SimuGen for recommending the second-best algorithm and determining which algorithm to discard in priority. SimuGen recommendations are also more stable than those of Proxy21k when considering only $c \in [2, 5]$ candidate algorithms out of the 6.

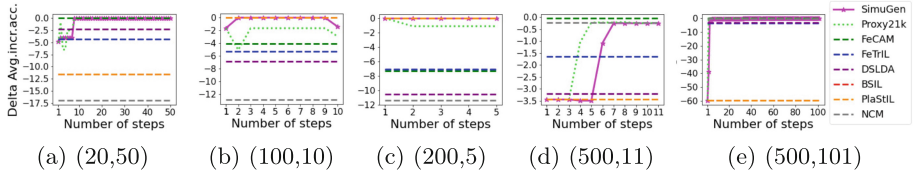


Fig. 3. Performance gap between the algorithm recommended by the oracle and by the proposed recommendation methods after simulating $t = 1, 2, \dots, T$ incremental steps, for scenarios of the form $(\text{Card}(P_1), T)$. Results are averaged over the three reference datasets (iN1K, iNat1k, Land1k).

7 Discussion

Performance. In this article, we have introduced a method for recommending a relevant DFCIL algorithm for applying it to a future data stream. We proposed two approaches for simulating a data stream from the same visual domain as the initial user dataset, one using generative models (SimuGen) and one using

existing databases (Proxy21k). An evaluation with six challenging incremental scenarios and three large-scale datasets covering various visual domains shows that *recommendations using SimuGen are close to that of an oracle*.

Usefulness. Our method facilitates the deployment of continual learning and requires little expertise on the part of the user. A heuristic consisting in choosing a fixed algorithm by default is quickly outdated and by far sub-optimal. We show that our method outperforms the fixed choice of any of the DFCIL algorithms we have tested. It can be adapted to new algorithms and use cases, and encourages the use of a wider range of algorithms proposed by the community. In addition to its usefulness in recommending DFCIL algorithms, this work provides a stress test for the algorithms studied. It shows that none is the best in all scenarios, and therefore underlines the importance of a comprehensive evaluation of CIL algorithms to understand their strengths and limitations.

Novelty. Our method exploits the knowledge encoded in a pre-trained LLM or knowledge base to simulate a data stream from the same domain as the user data. To the best of our knowledge, it is the first to take into account the semantic content of the stream to recommend a relevant CIL algorithm.

Further applications. *We have chosen the data-free (DFCIL) case because it is challenging and covers the deployment of continual learning in resource-constrained environments, such as embedded systems [13, 45].* We did not include algorithms relying on a memory buffer, as the comparison with DFCIL algorithms would have been unfair [2, 25]. Our approach can be applied to other areas of continual learning where variations in the relative performance of algorithms depending on the use case are observed too. This is the case of CIL with memory and task-incremental learning [2, 22]. It can also be adapted for domain-incremental learning, in which the set of classes is fixed, but the distribution of the classes changes. In addition, our approach is useful for comparing algorithms in constrained settings that are not taken into account in the main benchmarks.

Next, we discuss our method’s limitations and their potential mitigation.

Relevance of simulated data. (i) We observed that the new class names proposed by the LLM sometimes lacked diversity from one prompt to another, hence the multiple runs with diversified prompts. To improve the outputs of the LLM, methods to limit hallucinations or peculiar outputs as highlighted in [49] could be used. Another way of automatically cleaning the data could consist in checking the existence of the proposed class names against a knowledge base such as Wikidata. (ii) Stable Diffusion can generate large datasets flexibly, with control over the semantic content of the data, as opposed to web-crawled data. However, it might be challenged by specific visual domains that are not well covered by its training data. Nonetheless, domains with limited data could benefit from our algorithm recommendation approach too, e.g. in a few-shot CIL setting. We note that generative models are trained with increasingly large and diversified datasets, and this will increase their usability. (iii) Our Proxy21k approach is similar to web retrieval in that it uses ImageNet-21k to simulate data streams, a dataset collected from the web. The results show that Proxy21k’s performance is inferior to SimuGen’s for two datasets and equivalent for IN1K,

which is sampled from ImageNet-21k and for which there is therefore no domain shift. This underlines the need for specialized databases or models to simulate a stream close to the future data distribution.

Cost of data generation. The improved performance provided by recommendations using SimuGen is accompanied by an initial computational cost due to the use of generative models. In line with standard CIL practice [9, 12], we recall that we consider the first step to be offline and not incremental. Consequently, the use of large generative models as a preliminary step is not a limitation in the framework considered. The cost of data generation could be reduced using more efficient textual [48] and visual generation models [23]. Another option would be to use Proxy21k when the initial dataset is well covered by an existing visual dataset, and SimuGen when this is not the case.

Cost of recommendation. To reduce the cost of DFCIL experiments, a pre-selection of candidate algorithms can be applied, taking into account practical criteria such as the possibility of updating the model on the device, the latency of a model update, or the storage required. In addition, the training of poorly performing algorithms can be stopped early, as we propose to do with the "explore then prune" strategy (Δ_{gen}^{3+} and Δ_{proxy}^{3+} in Table 1). Figure 3 also shows that, in most cases, it is sufficient to run half of the simulation steps to obtain an accurate recommendation, as the ranking of the algorithms is stable at the end of the incremental process.

8 Conclusion

Despite intensive research in this field, no existing CIL algorithm performs best in all settings [2, 7]. It is thus necessary to recommend an algorithm for optimal deployment. Our work is the first to recommend an incremental learning algorithm based on a simulated data stream adapted to the semantic content of a user-defined scenario. We show that by leveraging generative models or an existing visual knowledge base, we can accurately recommend DFCIL algorithms in various visual domains and incremental settings. Our method could be extended to evaluate algorithms in other resource-constrained scenarios. We plan to experiment with other continual learning scenarios and study the impact of data stream structure and semantics on the performance of CIL algorithms.

Acknowledgements. This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council. We thank Marina Reyboz for her methodological advice. We also thank Paul Grimal and Michael Soumm for their advice on generative models.

References

1. Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: Conference on Computer Vision and Pattern Recognition. CVPR (2017)

2. Belouadah, E., Popescu, A., Kanellos, I.: A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Netw.* **135**, 38–54 (2021)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
4. Chrysakis, A., Moens, M.F.: Simulating task-free continual learning streams from existing datasets. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2515–2523 (2023)
5. Cossu, A., Ziosi, M., Lomonaco, V.: Sustainable artificial intelligence through continual learning. In: *International Conference on AI for People: Towards Sustainable AI, CAIP 2021, Bologna, Italy*. p. 103. European Alliance for Innovation (2021)
6. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: Imagenet: A large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, Florida, USA. pp. 248–255 (2009)
7. Feillet, E., Petit, G., Popescu, A., Reyboz, M., Hudelot, C.: Advisil - a class-incremental learning advisor. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 2400–2409 (2023)
8. French, R.M.: Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **3**(4), 128–135 (1999)
9. Goswami, D., Liu, Y., Twardowski, B., van de Weijer, J.: Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems* **36** (2024)
10. Harun, M.Y., Gallardo, J., Hayes, T.L., Kanan, C.: How efficient are today’s continual learning algorithms? In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2430–2435 (2023)
11. Hayes, T.L., Kafle, K., Shrestha, R., Acharya, M., Kanan, C.: Remind your neural network to prevent catastrophic forgetting. In: *European Conference on Computer Vision*. pp. 466–483. Springer (2020)
12. Hayes, T.L., Kanan, C.: Lifelong machine learning with deep streaming linear discriminant analysis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. pp. 220–221 (2020)
13. Hayes, T.L., Kanan, C.: Online continual learning for embedded devices. In: *Conference on Lifelong Learning Agents*. pp. 744–766. PMLR (2022)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Conference on Computer Vision and Pattern Recognition. CVPR* (2016)
15. Hemati, H., Cossu, A., Carta, A., Hurtado, J., Pellegrini, L., Bacciu, D., Lomonaco, V., Borth, D.: Class-incremental learning with repetition pp. arXiv–2301 (2023)
16. Hersche, M., Karunaratne, G., Cherubini, G., Benini, L., Sebastian, A., Rahimi, A.: Constrained few-shot class-incremental learning. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 9057–9067 (2022)
17. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. *CoRR* **abs/1503.02531** (2015)
18. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA*,. pp. 831–839 (2019)
19. Jodelet, Q., Liu, X., Murata, T.: Balanced softmax cross-entropy for incremental learning with and without memory. *Comput. Vis. Image Underst.* **225**, 103582 (2022)
20. Jodelet, Q., Liu, X., Phua, Y.J., Murata, T.: Class-incremental learning using diffusion model for distillation and replay. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 3425–3433 (2023)

21. Jung, D., Han, D., Bang, J., Song, H.: Generating instance-level prompts for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11847–11857 (2023)
22. Lee, K.Y., Zhong, Y., Wang, Y.X.: Do pre-trained models benefit equally in continual learning? In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 6485–6493 (January 2023)
23. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems* **36** (2024)
24. Li, Z., Hoiem, D.: Learning without forgetting. In: European Conference on Computer Vision. ECCV (2016)
25. 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 TPAMI* **45**(5), 5513–5533 (2022)
26. Miller, G.A.: Wordnet: a lexical database for english. *Commun. ACM* **38**(11), 39–41 (1995)
27. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: IEEE ICCV. pp. 3476–3485 (2017)
28. Petit, G., Popescu, A., Belouadah, E., Picard, D., Delezoide, B.: Plastil: Plastic and stable exemplar-free class-incremental learning. In: Conference on Lifelong Learning Agents. pp. 399–414. PMLR (2023)
29. Petit, G., Popescu, A., Schindler, H., Picard, D., Delezoide, B.: Fetril: Feature translation for exemplar-free class-incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3911–3920 (January 2023)
30. Petit, G., Soumm, M., Feillet, E., Popescu, A., Delezoide, B., Picard, D., Hudelot, C.: An analysis of initial training strategies for exemplar-free class-incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1837–1847 (2024)
31. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* **21**(1), 5485–5551 (2020)
32. Rebuffi, S., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Conference on Computer Vision and Pattern Recognition. CVPR (2017)
33. Ring, M.B.: Continual Learning in Reinforcement Environments. Ph.D. thesis, University of Texas at Austin, USA (1994), uMI Order No. GAX95-06083
34. 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. pp. 10684–10695 (2022)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **115**(3), 211–252 (2015)
36. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint [arXiv:1606.04671](https://arxiv.org/abs/1606.04671) (2016)
37. Saryildiz, M.B., Alahari, K., Larlus, D., Kalantidis, Y.: Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In: IEEE Conference on Computer Vision and Pattern Recognition (2023)
38. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. *Advances in neural information processing systems* **30** (2017)

39. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
40. Tabassum, A., Erbad, A., Mohamed, A., Guizani, M.: Privacy-preserving distributed ids using incremental learning for iot health systems. *IEEE Access* **9**, 14271–14283 (2021)
41. Tian, Y., Fan, L., Isola, P., Chang, H., Krishnan, D.: Stablerep: Synthetic images from text-to-image models make strong visual representation learners. arXiv preprint [arXiv:2306.00984](https://arxiv.org/abs/2306.00984) (2023)
42. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint [arXiv:2302.13971](https://arxiv.org/abs/2302.13971) (2023)
43. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
44. van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three types of incremental learning. *Nature Machine Intelligence* **4**(12), 1185–1197 (2022)
45. Verwimp, E., Ben-David, S., Bethge, M., Cossu, A., Gepperth, A., Hayes, T.L., Hüllermeier, E., Kanan, C., Kudithipudi, D., Lampert, C.H., et al.: Continual learning: Applications and the road forward. [arXiv:2311.11908](https://arxiv.org/abs/2311.11908) (2023)
46. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024)
47. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
48. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., Han, S.: Smoothquant: Accurate and efficient post-training quantization for large language models. In: International Conference on Machine Learning. pp. 38087–38099. PMLR (2023)
49. Ye, H., Liu, T., Zhang, A., Hua, W., Jia, W.: Cognitive mirage: A review of hallucinations in large language models. arXiv preprint [arXiv:2309.06794](https://arxiv.org/abs/2309.06794) (2023)
50. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., van de Weijer, J.: Semantic drift compensation for class-incremental learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020. pp. 6980–6989. IEEE (2020)
51. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)
52. Zhu, K., Zhai, W., Cao, Y., Luo, J., Zha, Z.J.: Self-sustaining representation expansion for non-exemplar class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9296–9305 (2022)



Incremental Object 6D Pose Estimation

Long Tian^{1,2(✉)}, Amelia Sorrenti³, Yik Lung Pang¹, Giovanni Bellitto³,
Simone Palazzo³, Concetto Spampinato³, and Changjae Oh¹

¹ Queen Mary University of London, London, United Kingdom
long.tian@qmul.ac.uk

² Southwest Jiaotong University, Chengdu, China

³ University of Catania, Catania, Italy

Abstract. We present a novel setting for 6D object pose estimation, where a model progressively adapts its parameters to estimate the pose of new objects without forgetting. This capability is crucial for real-world applications, particularly in scenarios where a deployed model must accommodate new objects while mitigating the risk of forgetting previously seen objects. To tackle this challenge, we propose a replay-based incremental learning technique designed to retain key information about previously seen objects when the model is exposed to a new one. Our approach relies on a memory buffer comprising keyframes of previously encountered objects, serving to regularize the model parameters based on past experiences while allowing for the update of model features to perform pose estimation on new objects. We validate the effectiveness of our method on the standard Linemod and YCB-Video datasets, demonstrating how our method surpasses baseline approaches in incremental learning at the task at hand. The project website is available at: <https://qm-ipalab.github.io/ILPose>.

Keywords: 6D Pose Estimation · Incremental Learning · Elastic Weight Consolidation

1 Introduction

6D pose estimation, i.e., the prediction of the 3D position and 3D orientation of a target object from images, is a fundamental problem in computer vision, with wide applications ranging from autonomous driving to virtual/augmented reality to robotic grasping. Supervised methods for 6D pose estimation typically rely on establishing keypoint correspondences between CAD models and input RGB images [28], RGB-D images [11, 37], or directly regress pose using input RGB-D images [38, 40]. Depending on the underlying assumptions, 6D pose estimation can be performed at either the instance level [11, 28] or the category level [37, 40].

However, the majority of these approaches are trained on large datasets in an offline setting, assuming that the training and test sets are independently and identically distributed (i.i.d.) and that all target objects are available for training at the same time. This assumption is mostly impractical in real-world

scenarios where a deployed model must incrementally adapt to new data. In such contexts, existing methods struggle to accurately estimate the 6D pose for unseen objects, whose feature distribution differs from those encountered in the training set. Moreover, the conventional practice of retraining the model with the entire dataset, whenever a new object is introduced, is unfeasible and not efficient: retraining the model demands significant computational resources and time, and storing and processing large datasets may overwhelm robots with limited memory capacities.

To address the limitation of offline retraining, one-shot [10,32] and few-shot [12] methods leverage annotated support views of new objects, establishing correspondences between these views and the query view for pose estimation. These methods relax the constraint for high-fidelity object models, but necessitate training on specific instances or categories. Test-time adaptation methods [18,19] attempt to bridge the gap between the training and test sets by adapting a pre-trained model to new objects encountered during testing, either through supervised or unsupervised training.

However, adapting the model only using data from new objects leads to *catastrophic forgetting*, a well-known phenomenon observed when a model is sequentially trained on different experiences or *tasks* [26]. In such scenarios, while the model effectively learns new objects, it tends to forget previously encountered ones after adaptation, resulting in diminished performance on past objects, as illustrated in Figure 1. This challenge arises due to significant changes in the model parameters during adaptation to new objects, without adequately retaining the knowledge acquired from previously encountered objects.

Incremental learning has emerged as a dedicated branch of machine learning to address this challenge by balancing between *plasticity*, i.e., adaptation to new information, and *stability*, i.e., ensuring that previously learned knowledge remains relevant and accessible over time. Existing works predominantly deal with image classification, with relatively little attention devoted to other domains such as robotic perception [20], reinforcement learning [1,41], and natural language processing [14]. To our knowledge, the exploration of incremental learning for 6D pose estimation has not been investigated. Accordingly, in this paper, we introduce a novel framework for learning the 6D pose of objects, specifically designed for scenarios where a model can adapt incrementally to new

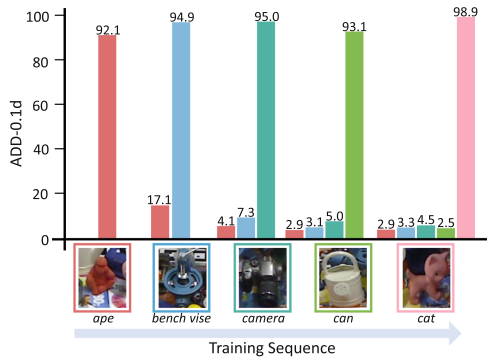


Fig. 1. Motivation. Performance degradation during sequential training on five objects of the Linemod dataset. Each color bar corresponds to a specific object. For example, pose estimation accuracy for *ape* (in red) drops from 92.1% at task 1 to 2.9% at task 5.

objects without losing knowledge of previously encountered ones. Our approach trains a model to estimate the 6D pose on a subset of objects (a *task*) at a time, selectively storing keyframes of objects from the current task in a memory buffer. At each new task, the model keeps learning by combining data from new objects and the memory buffer, which is simultaneously updated to accommodate keyframes of the new task objects. Throughout this incremental learning process, we introduce a parameter regularization method aimed at adaptively adjusting the model parameters to retain those essential for accurately estimating the pose of objects stored in the memory buffer. Furthermore, we propose a strategy to alleviate overfitting on the objects contained within the memory buffer. Overall, the main contributions of this work are:

- We present a novel setting for 6D object pose estimation, wherein the model progressively learns to estimate the 6D pose of new objects.
- We introduce a replay-based incremental learning method that prioritizes adaptability to new objects while ensuring retention of knowledge about previously seen ones.
- We validate our method by comparing with existing incremental learning baselines on the Linemod [13] and YCB-Video [42] datasets.

2 Related work

2.1 6D pose estimation

Methods for 6D pose estimation can be classified as either *instance-level* or *category-level*, depending on their generalization capabilities. *Instance-level* methods [28, 38, 45] assume that target object CAD models are available, and focus on estimating the 6D pose of specific objects. *Category-level* methods [23, 39, 40] assume that category information is available, and build models to learn the category-specific representation of object appearance and shapes, enabling 6D pose estimation within the same category. However, since instance-level methods require high-fidelity CAD models for each object of interest and category-level methods require prior knowledge of the target category, neither can adequately handle new objects that are not shown in the training set.

To address the challenge of achieving 6D pose estimation for new objects, one-shot [10, 32] and few-shot [12] 6D pose estimation methods have been proposed. These methods annotate one or several views of the new object as support views and then match keypoints between the support views and the query scene for 6D pose estimation [8, 35]. These methods avoid using CAD models but require training a separate model for each object. Test-time adaptation is another approach to estimate the 6D pose of new objects [33], which is used to enhance the performance of the model when there is a distribution shift between known and new objects. These approaches [19, 34] pre-train a model on known objects in a supervised manner, and then adapt the pre-trained model to new objects in a supervised or unsupervised manner. However, the model would forget previous knowledge after adaptation.

2.2 Incremental learning

Incremental learning refers to the ability of a model to learn from new data without suffering from the *catastrophic forgetting* of the previously learned knowledge [26]. Most existing works on incremental learning have been designed for classification tasks and reinforcement learning, and can be broadly categorized into three directions: architectural-based [25, 31, 43], regularization-based [17, 22, 44], and rehearsal-based methods [2, 3, 5]. Hybrid methodologies, blending the strengths of various approaches, have also been proposed [4].

Architecture-based methods dynamically adjust the structure of the neural network, either by pruning unnecessary parts of the network [25, 31] or by introducing new parameters to handle the new incoming tasks [30, 43]. This category also includes mask-based methods, where a mask is learned for individual weights or groups of weights, enabling the selective freezing or constraint of specific parameters [15, 27]. Regularization-based methods leverage regularization terms to mitigate significant changes in important weights, penalizing updates on these weights to preserve previously learned knowledge [22, 44]. Among these methods, Elastic Weight Consolidation (EWC) [17] stands out as an effective approach, through the use of a Fisher information matrix to identify the importance of each trainable parameter. Rehearsal-based methods retain samples from previous tasks in a memory buffer and revisit these samples while adapting to new data, mitigating forgetting by preserving previously learned knowledge [3, 5]. The replay strategy, which has been demonstrated to be effective in the incremental learning scenario, is often combined with a regularization strategy on logits sampled over the optimization trajectory [4]. In this work, we propose a combination of rehearsal-based and regularization-based methods, to develop a comprehensive incremental learning framework for the 6D pose estimation task.

3 Method

We define the proposed incremental object 6D pose estimation as a task-incremental learning problem, where a network \mathcal{F} undergoes training on a sequence of T tasks $\{\tau_1, \dots, \tau_T\}$ to estimate the 6D pose of multiple objects. Each task involves learning the 6D pose of a specific subset of objects from a set $O = \{o_1, \dots, o_n\}$ with annotated poses. Formally, τ_i represents the subset of objects for the i -th task, such that $\tau_i \subset O$, $\tau_i \cap \tau_j = \emptyset$, and $\bigcup \tau_i = O$.

We design a network \mathcal{F} that takes as input a pair $d = (v, s)$, where v is the RGB-D image and s is the segmentation mask for the target object. The network \mathcal{F} directly regresses the object 6D pose between the camera and object coordinate system, represented as $p = \mathcal{F}(\theta|v, s)$, with θ being the trainable parameters. We represent 6D pose as a homogeneous transformation matrix p that consists of a 3D rotation $R \in SO(3)$ and a 3D translation $t \in \mathbb{R}^3$.

Figure 2 shows the overview of our method. During each task τ_i , we employ a selection process to identify k keyframes, maximizing multi-view diversity. These selected keyframes, along with their corresponding ground truth poses, are stored in a memory buffer \mathcal{M} . At each task, the model \mathcal{F} is trained to estimate the

pose of the task’s objects; additionally, starting from task τ_2 , we include data from the memory buffer \mathcal{M} in the training process.

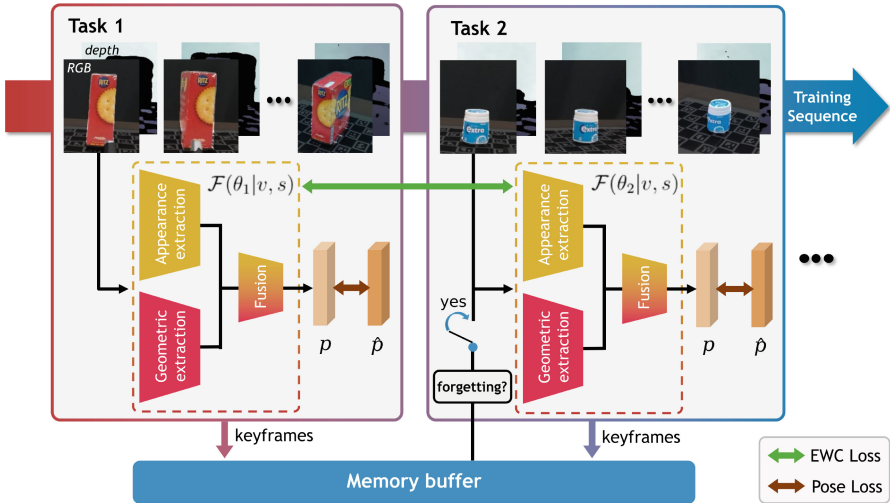


Fig. 2. Incremental object 6D pose estimation framework. Our model takes an RGB-D frame and its target object mask as input for 6D pose estimation. The trained model is sequentially adapted to new objects by using data from both the new object and the memory buffer, where keyframes of previously seen objects are stored.

3.1 Object pose estimation

We employ a model that directly estimates the 6D pose information from input images, leveraging both color and geometric features to address pose estimation challenges for objects with low texture or similar colors. Given an RGB-D image as input, we crop the RGB image and its corresponding depth image using the segmentation mask of the target object. This ensures that the model focuses solely on the region containing the target object.

For RGB feature extraction, we utilize a ResNet-18 [9] as an encoder to extract appearance features from the cropped image. Simultaneously, we extract geometric features by lifting the target object’s point cloud from the cropped depth images using camera intrinsic parameters. We employ PointNet [29] to process the point cloud and obtain geometric features for each 3D point. We randomly select 500 points with the corresponding pixels in RGB image, fuse their extracted appearance and geometric features, and obtain a pixel-wise feature representation using DenseFusion [38]. Finally, we regress the 6D pose of the target object by leveraging the pixel-wise feature representation.

The estimation model is trained by minimizing a pose loss \mathcal{L}_{pose} that encourages the estimation of accurate pose information. We use the Average Distance of

Model (ADD) scores [13] as \mathcal{L}_{pose} . It measures the distance between points transferred using the predicted 6D pose and points transferred through the ground truth pose, expressed as:

$$\mathcal{L}_{pose} = \frac{1}{T} \sum_i \left\| (Rx_i + t) - (\hat{R}x_i + \hat{t}) \right\|^2, \quad (1)$$

where x_i represents the i^{th} 3D point from the object CAD model (with T overall number of points), \hat{R} and \hat{t} denote the rotation and translation annotations and R and t the estimated rotation and translation values.

3.2 Memory-based incremental learning

We design a replay-based strategy for incremental object 6D pose estimation. This strategy consists of two key components: *keyframe selection* and *adaptive parameter regularization*.

Keyframe selection. The selection of samples from past tasks is a crucial step, as these samples directly influence the model’s robustness on previous tasks following adaptation to new ones. The memory buffer contains frames that must effectively provide exhaustive visual appearance information about seen objects. To achieve this, we propose a selection strategy based on the Scale-Invariant Feature Transform (SIFT) algorithm [24].

SIFT is primarily employed to detect and describe local features in images by identifying distinctive matching keypoints in two images and estimating the transformation between them. We utilize this transformation as a measure of the view changes between sequential frames. Hence, we leverage SIFT to quantify the changes in view across consecutive frames and select *keyframes* that maximize multi-view diversity while minimizing redundancy.

Formally, given a sequence of frames $\{v_1, \dots, v_n\}$ depicting an arbitrary object $o \in O$, we select the initial frame v_1 . Subsequently, we compare the other $n - 1$ frames with v_1 using SIFT to find matched keypoints for transformation calculation, yielding a set of scores $\mathbf{S} = \{s_{1,j}, \text{ with } j = 2 \dots n\}$ with $s_{1,j}$ measuring the Euclidean norm of translation and rotation angle, indicating the extent of view changes. The k frames with the highest scores are identified as *keyframes* and added to the memory buffer \mathcal{M} . The number of keyframes in the buffer is equal for all objects.

Adaptive parameter regularization. While adapting the current model to a new task, knowledge of past tasks may be overwritten or altered, leading to catastrophic forgetting. We address the forgetting problem in our incremental 6D pose estimation by adaptively regularizing the parameters using a variant of Elastic Weight Consolidation (EWC) [17]. In particular, while the common online EWC implementation computes the Fisher information matrix \mathcal{G} at the end of each task, on the task’s data only, and updates it through exponential

Algorithm 1. Incremental object 6D pose estimation for each task

```

1: Input: data from both the task data distribution  $\mathcal{D}_n$  and the memory buffer  $\mathcal{M}$ 
2: for all epochs do
3:   for  $v_i, s_i \in \mathcal{D}_n$  do                                     ▷ training of the model on  $\mathcal{D}_n$ 
4:      $p_i \leftarrow \mathcal{F}(\theta|v_i, s_i)$ 
5:      $Loss \leftarrow Loss + \mathcal{L}_{pose}(p_i) + \lambda\mathcal{L}_{EWC}(p_i)$ 
6:   end for
7:   for  $v_j, s_j \in \mathcal{M}$  do                                     ▷ evaluation of the model on  $\mathcal{M}$ 
8:      $p_j \leftarrow \mathcal{F}(\theta|v_j, s_j)$ 
9:      $Loss_{eval} \leftarrow Loss_{eval} + \mathcal{L}_{pose}(p_j)$ 
10:  end for
11:  if  $(Loss_{eval} / |\mathcal{M}|) > \delta$  then
12:    for  $v_j, s_j \in \mathcal{M}$  do                                     ▷ training of the model on  $\mathcal{M}$ 
13:       $p_j \leftarrow \mathcal{F}(\theta|v_j, s_j)$ 
14:       $Loss \leftarrow Loss + \mathcal{L}_{pose}(p_j) + \lambda\mathcal{L}_{EWC}(p_j)$ 
15:    end for
16:  end if
17:  update Fisher information matrix  $\mathcal{G}$                                ▷ using data from  $\mathcal{M}$ 
18:  update  $\theta$  based on  $\frac{\partial Loss}{\partial \theta}$ 
19: end for
20: update  $\mathcal{M}$  with data from the current task data distribution  $\mathcal{D}_n$ 

```

moving average, we compute it using all samples from the buffer \mathcal{M} , through the second-order derivative:

$$\mathcal{G} = \mathbb{E}\left[-\frac{\partial^2}{\partial \theta_{\tau_i}^2} \ell(p_m | \theta_{\tau_i})\right], \quad (2)$$

where p_m is the ground truth for the m -th sample in the buffer and $\ell(p_m | \theta_{\tau_i})$ is the corresponding log-likelihood [21]. As the Fisher information matrix is related to the importance of each parameter, we add a regularization loss to penalize the change of each parameter according to its importance:

$$\mathcal{L}_{EWC} = \sum_j \mathcal{G}(\theta_{\tau_{i-1}}^{(j)} - \theta_{\tau_i}^{(j)})^2, \quad (3)$$

where $\theta_{\tau_{i-1}}^{(j)}$ represents the j^{th} parameter of the weights at task τ_{i-1} and $\theta_{\tau_i}^{(j)}$ is the same parameter of the model at task τ_i .

3.3 Optimization

Our learning strategy foresees that the pose estimation model is trained on the first task in a supervised manner, using only Eq. 1. For subsequent tasks, the model is trained by minimizing the overall loss \mathcal{L}_{adp} :

$$\mathcal{L}_{adp} = \mathcal{L}_{pose} + \lambda\mathcal{L}_{EWC}, \quad (4)$$

where \mathcal{L}_{pose} is computed for all samples of the new task, while \mathcal{L}_{EWC} only on the samples present in the replay buffer \mathcal{M} . However, a common challenge in replay-based incremental learning methods is the tendency to overfit the buffer data [7]. This occurs as the model repeatedly learns from the buffer, potentially hindering generalization to previous tasks and causing loss of previously learned knowledge. Additionally, using a buffer directly without additional training strategies, often results in a polarization towards the new object data, whose effect is controlled by the hyperparameter λ .

To mitigate this issue, we propose a novel adaptation strategy, adding a loss-gating mechanism. We exclusively use data from the new object to enable rapid adjustment of the network’s trainable parameters to suit the characteristics of the new object. During training, after each epoch, we evaluate network performance on memory buffer data to detect any significant drop in performance on previously seen objects. If the loss value (i.e., the ADD value obtained by Eq.1) exceeds a predefined threshold, δ , indicating a decline in performance, we incorporate memory buffer data for training using Eq. 4. Conversely, if the loss value remains below the threshold, signifying retention of previously encountered objects, we refrain from adding memory buffer data to training to prevent overfitting. The pseudocode of the procedure performed for each task is shown in Alg. 1.

4 Experiments

4.1 Metrics

We consider three metrics widely used in 6D pose estimation problems:

- **ADD-0.1d** (\uparrow): percentage of correct poses. The estimated pose is considered to be correct if the ADD distance is less than 10% of the object diameter.
- **R_{err}** (\downarrow): mean rotation error in degrees, which measures the average angle between the predicted rotation and the ground truth rotation.
- **T_{err}** (\downarrow): mean translation error in centimeters, which is used to measure the average Euclidean distance between the predicted translation and the ground truth.

We assess the effectiveness of our method by evaluating its overall performance on both past and present tasks, encompassing all objects encountered so far. To accomplish this, we report the performance of the model in terms of *Final Average* metrics [5]: $ADD-0.1d^{FA}(\uparrow)$, $R_{err}^{FA}(\downarrow)$ and $T_{err}^{FA}(\downarrow)$. These metrics represent the average performance measures over all objects encountered after the last task of the sequence. Let ψ_i^j denote the value of an arbitrary metric at the end of task j computed on the test set of task τ_i (with $i \leq j$), the *Final Average* is defined as:

$$\Psi^{FA} = \frac{1}{T} \sum_{i=1}^T \psi_i^T, \quad (5)$$

with T representing the total number of tasks.



Fig. 3. Visualization of 6D pose. The results on (top) Linemod [13] and (bottom) YCB-Video [42]. The green 3D bounding boxes are obtained based on the ground truth 6D pose, while the red ones are generated from the model prediction after the incremental adaptation to the final object.

4.2 Baselines

We compare our method, *ILPose*, with a selection of other approaches broadly inspired by existing state-of-the-art incremental learning methods. Most of these methods are originally designed for the classification task, so some changes have been made to adapt them to the 6D pose estimation task.

Multi-Encoder: This strategy draws inspiration from a popular architectural method [30], in which a distinct replica of the entire backbone is dedicated to each task. While this strategy inherently prevents forgetting, its primary drawback lies in the linear increase of memory requirements with the number of tasks. To balance efficiency and memory footprint, our implementation allocates a separate encoder module for each task, while the rest of the network remains shared across all tasks. When a new task begins, a new encoder replica is instantiated, initialized with the same weights as its predecessor, and adopted during the subsequent training session, while all the other encoders are inactive. During inference, an object identifier is used to select the appropriate encoder.

Self-Distillation: Inspired by [22], this method applies functional regularization via self-distillation between the in-training model and a previous snapshot stored in the buffer. More specifically, at the end of task τ_{i-1} , we store the pixel-wise features $h_{\tau_{i-1}}$ for each buffer sample $m \in \mathcal{M}$. In the next task τ_i , we incorporate an additional loss \mathcal{L}_{SD} to mitigate potential degradation the learned representations up to task τ_{i-1} . At each training step of the current task τ_i , we sample an image v from the training stream, and a m image randomly selected from the buffer, and we optimize the network by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{pose}^{(v)} + \mathcal{L}_{pose}^{(m)} + \alpha \mathcal{L}_{SD}(h_{\tau_{i-1}}^{(m)}, h_{\tau_i}^{(m)}), \quad (6)$$

where α is a weighting factor between the three loss terms, and \mathcal{L}_{SD} is the Mean Squared Error Loss.

Hybrid: This approach combines the strengths of the first two methods, leveraging the advantages of both techniques. It employs a separate encoder for

each new object, coupled with the additional loss \mathcal{L}_{SD} to preserve the knowledge associated with the previous state of the model.

vanilla-EWC [17]: In this buffer-free method, the model continually adapts to new objects by updating trainable parameters under the penalty of the EWC term applied to important parameter changes. In this case, the loss-gating mechanism is not being added.

Joint & Fine-tune: To provide a more exhaustive understanding of our findings, we also include the scenario where a model is trained jointly on all objects together (referred to as *Joint*) in a conventional, non-incremental fashion. In addition, we present the results by training the model sequentially on each task without implementing any measures to contrast forgetting (referred to as *Fine-tune*). These two results can be viewed as upper and lower bounds, respectively.

Table 1. 6D pose estimation results on Linemod [13].

Method	ADD-0.1d ^{FA} (↑)	R _{err} ^{FA} (↓)	T _{err} ^{FA} (↓)
Joint	90.3±0.7	7.7±1.1	0.7±0.2
Fine-tune	18.1±1.9	50.4±6.4	3.1±0.3
vanilla-EWC	36.6±2.5	27.7±1.7	1.7±0.1

	30 keyframes			50 keyframes		
	ADD-0.1d ^{FA} (↑)	R _{err} ^{FA} (↓)	T _{err} ^{FA} (↓)	ADD-0.1d ^{FA} (↑)	R _{err} ^{FA} (↓)	T _{err} ^{FA} (↓)
Multi-Encoder	29.6±3.1	30.3±2.0	9.2±1.2	39.9±4.8	23.8±2.9	8.4±1.0
Self-Distillation	39.3±3.5	31.9±3.3	6.4±0.9	44.6±4.4	24.3±2.1	8.7±0.9
Hybrid	34.4±4.7	29.1±2.8	8.7±1.3	50.9±6.7	21.7±3.3	7.7±1.1
ILPose	58.4±3.0	14.4±1.1	1.6±0.1	67.5±1.1	10.8±1.8	1.3±0.1

4.3 Experimental results

Setup. All models are trained according to a standard incremental learning protocol [36]. When training on a given task, only images corresponding to that task are used, with the exception of k samples for each previous object stored in the buffer (if the method permits). To ensure a fair comparison between different methods, all the networks are trained using the Adam [16] optimizer for 300 epochs per task. We select $\lambda = 2$ to balance the performance on a new task and retention of previous knowledge, while the threshold δ is determined based on the object diameter d , defined as $\delta = 0.1d$.

For each method, the model is trained on a sequence of 5 objects from either the Linemod dataset [13] (*ape*, *bench vise*, *camera*, *can* and *cat*) or the YCB-Video dataset [6] (*master chef can*, *cracker box*, *sugar box*, *tomato soup can* and *mustard bottle*); for both datasets, each task is associated to a single object, i.e., $\tau_i = o_i$. The evaluation of the methods equipped with a buffer was performed with two different buffer size settings, i.e. storing $k = 30$ and $k = 50$ images

per object. Results are presented as the mean and standard deviation over three different runs.

Results. Table 1 and Table 2 show that ILPose outperforms all baselines on the Linemod and YCB-Video datasets, providing a better trade-off between forgetting and final performance. In the tables, we also compare the impact of the number of keyframes on final results. Qualitative results are shown in Figure 3.

Although vanilla-EWC does not leverage frames from previous tasks, it yields results comparable to buffer-based methods, particularly in the case of $k = 30$. This serves as a direct baseline for our ILPose, suggesting that regularization based on the Fisher information matrix effectively addresses our task.

Among buffer-based methods, the architectural approach utilizing multiple encoders performs poorly across all cases. Employing separate encoders for each task seems ineffective, as the encoder’s internal representation remains closely tied to the inherent features of the object in question. Consequently, the shared part of the network struggles to exploit features extracted by different encoders. This phenomenon may also explain the results obtained with the Self-Distillation method, which, despite using a single encoder, better preserves the model’s internal representation due to the additional loss. Clearly, when the network retains its weights to ensure that current pixel-wise features resemble those extracted previously, it suffers less from forgetting.

We evaluate the impact of catastrophic forgetting on the 6D pose estimation task by comparing the performance metrics at the end of each task between two scenarios: one with Fine-tune (lacking countermeasures to reduce forgetting) and the other with our proposed solution. Figure 5 illustrates this comparison, demonstrating a stark difference in performance. While the former shows a significant drop, our approach notably mitigates this decline.

Table 2. 6D pose estimation results on YCB-Video [42].

Method	ADD-0.1d ^{FA} (↑)	R_{err}^{FA} (↓)	T_{err}^{FA} (↓)
Joint	96.2±0.9	5.7±0.5	0.5±0.1
Fine-tune	27.8±2.1	52.4±2.7	3.9±0.5
vanilla-EWC	46.8±1.4	31.2±1.6	2.5±0.3
	30 keyframes		
	ADD-0.1d ^{FA} (↑)	R_{err}^{FA} (↓)	T_{err}^{FA} (↓)
Multi-Encoder	35.6±1.1	26.1±0.8	8.2±1.4
Self-Distillation	39.4±2.5	29.8±1.3	3.2±2.1
Hybrid	36.8±2.9	25.2±2.7	5.9±2.3
ILPose	63.5±2.8	15.2±0.9	1.6±0.2
	50 keyframes		
	ADD-0.1d ^{FA} (↑)	R_{err}^{FA} (↓)	T_{err}^{FA} (↓)
Multi-Encoder	38.9±1.4	23.9±2.2	6.3±0.8
Self-Distillation	45.1±2.7	25.1±1.0	4.4±1.5
Hybrid	49.2±3.9	21.7±3.1	6.7±1.4
ILPose	79.2±1.8	10.2±0.7	1.2±0.1

Table 3. Effect of regularization. We compare our original model, ILPose, which employs EWC, with the L_2 -based which utilizes L_2 regularization, while keeping all other settings remain unchanged.

Method	30 keyframes			50 keyframes		
	ADD-0.1d ^{FA} (↑)	R_{err}^{FA} (↓)	T_{err}^{FA} (↓)	ADD-0.1d ^{FA} (↑)	R_{err}^{FA} (↓)	T_{err}^{FA} (↓)
L_2 -based	42.7±1.4	24.7±2.1	1.5±0.1	52.4±2.1	21.1±1.4	1.3±0.2
ILPose	58.4±3.0	14.4±1.1	1.6±0.1	67.5±1.1	10.8±1.8	1.3±0.1

4.4 Ablation study

Impact of EWC term. To determine how parameter regularization can assist ILPose in preventing forgetting, we sequentially train the model on new objects. While the model may achieve satisfactory results on the most recently encountered object, significant forgetting can occur on the older objects. The 6D pose results for each encountered object are illustrated in Figure 5 (second row), leading to unsatisfactory average results across all encountered objects, as displayed in Table 1 (Fine-tune). For a fair comparison, we conduct another experiment that replaces EWC with L_2 regularization term while keeping other settings the same as ILPose. Unlike EWC, L_2 regularization penalizes all trainable parameter changes. Consequently, the model using L_2 regularization tends to retain the first encountered object as fewer parameters can be updated to fit newly encountered objects. The results are displayed in Table 3. EWC regularization employs the Fisher matrix that measures the important parameters for past encountered objects, which allows the model to be more flexible in handling new objects by penalizing only significant changes.

Impact of keyframes selection.

To assess the impact of keyframes selection on the results, we compared two different strategies for filling the memory buffer. In the first strategy, the keyframes were selected from the training set without taking into account any discriminative criteria, i.e. *random selection*. In contrast, the other selection strategy relies on the SIFT algorithm. As detailed in Section 3.2, the *SIFT-based selection* aims at capturing diverse and informative keyframes. For each approach, we conducted experiments using the same task order and training configuration. Figure 4 shows a

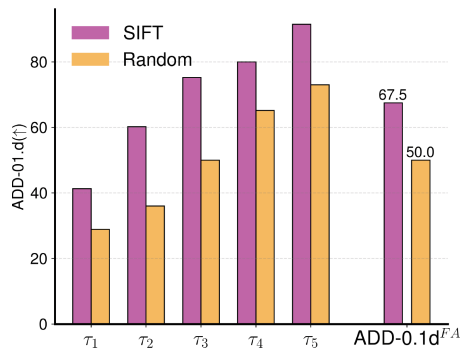


Fig. 4. Comparison of the impact of different keyframes selection strategies. Per-task ADD-0.1d computed at the end of the training on the Linemod dataset. On the right, we report the ADD-0.1d^{FA}(↑).

comparison of the results in terms of $\text{ADD-0.1d}^{FA}(\uparrow)$. These results demonstrate the limitations of random selection in losing crucial features related to different poses, highlighting the effectiveness of SIFT-based evaluation in preserving such information.

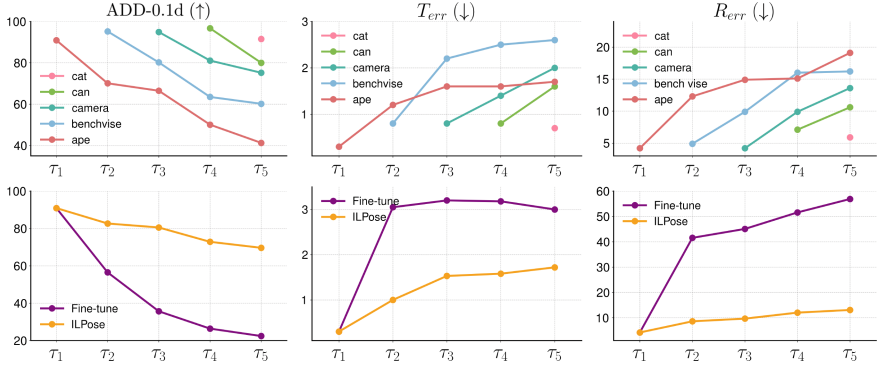


Fig. 5. Evaluation metrics for ILPose on Linemod [13]. First row: per-task $\text{ADD-0.1d}(\uparrow)$, $R_{err}(\downarrow)$ and $T_{err}(\downarrow)$ computed after training on each task. Second row: comparison between average metrics of ILPose and the Fine-tune model.

5 Conclusion

We presented a new setting for 6D pose estimation and introduced ILPose, addressing the challenge of incremental learning in 6D object pose estimation. ILPose leverages a memory buffer to retain the knowledge of previously seen objects while adapting to new ones and adaptively regularizes the model parameters to ensure keeping the previously acquired knowledge. Through extensive experiments on the Linemod and the YCB-Video datasets, we demonstrate that ILPose outperforms existing baselines in incremental 6D pose estimation, showcasing the effectiveness of our approach in real-world scenarios.

Acknowledgements. Amelia Sorrenti is a PhD student enrolled in the National PhD in Artificial Intelligence, cycle XXXVIII, course on Health and life sciences, organized by Università Campus Bio-Medico di Roma. G. Bellitto and C. Spampinato acknowledge financial support from PNRR MUR project PE0000013-FAIR.

References

1. Abel, D., Barreto, A., Van Roy, B., Precup, D., van Hasselt, H.P., Singh, S.: A definition of continual reinforcement learning. *NeurIPS* (2024)
2. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. *NeurIPS* (2019)

3. Arani, E., Sarfraz, F., Zonooz, B.: Learning fast, learning slow: A general continual learning method based on complementary learning system. arXiv preprint (2022)
4. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark Experience for General Continual Learning: a Strong, Simple Baseline. In, NeurIPS (2020)
5. Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., Belilovsky, E.: New Insights on Reducing Abrupt Representation Change in Online Continual Learning. In: ICLR (2022)
6. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The YCB object and model set: Towards common benchmarks for manipulation research. In: ICAR (2015)
7. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE TPAMI **44**(7), 3366–3385 (2021)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
10. He, X., Sun, J., Wang, Y., Huang, D., Bao, H., Zhou, X.: OnePose++: Keypoint-free one-shot object pose estimation without CAD models. NeurIPS (2022)
11. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation. In: CVPR (2020)
12. He, Y., Wang, Y., Fan, H., Sun, J., Chen, Q.: FS6D: Few-shot 6D pose estimation of novel objects. In: CVPR (2022)
13. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: ICCV (2011)
14. Hu, H., Sener, O., Sha, F., Koltun, V.: Drinking from a firehose: Continual learning with web-scale natural language. IEEE TPAMI **45**(5), 5684–5696 (2022)
15. Kang, H., Mina, R.J.L., Madjid, S.R.H., Yoon, J., Hasegawa-Johnson, M., Hwang, S.J., Yoo, C.D.: Forget-free continual learning with winning subnetworks. In: ICML (2022)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint (2014)
17. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences (2017)
18. Lee, T., Lee, B.U., Shin, I., Choe, J., Shin, U., Kweon, I.S., Yoon, K.J.: UDA-COPE: Unsupervised domain adaptation for category-level object pose estimation. In: CVPR (2022)
19. Lee, T., Tremblay, J., Blukis, V., Wen, B., Lee, B.U., Shin, I., Birchfield, S., Kweon, I.S., Yoon, K.J.: TTA-COPE: Test-time adaptation for category-level object pose estimation. In: CVPR (2023)
20. Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Díaz-Rodríguez, N.: Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. Information Fusion **58**, 52–68 (2020)
21. Li, Y., Zhang, R., Lu, J., Shechtman, E.: Few-shot image generation with elastic weight consolidation. arXiv preprint [arXiv:2012.02780](https://arxiv.org/abs/2012.02780) (2020)

22. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE TPAMI* **40**(12), 2935–2947 (2017)
23. Lin, J., Wei, Z., Zhang, Y., Jia, K.: Vi-net: Boosting category-level 6D object pose estimation via learning decoupled rotations on the spherical representations. In: *ICCV* (2023)
24. Lowe, D.G.: Object recognition from local scale-invariant features. In: *ICCV* (1999)
25. Mallya, A., Lazebnik, S.: PackNet: Adding multiple tasks to a single network by iterative pruning. In: *CVPR* (2018)
26. 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)
27. Miao, Z., Wang, Z., Chen, W., Qiu, Q.: Continual learning with filter atom swapping. In: *ICLR* (2021)
28. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: Pixel-wise voting network for 6DoF pose estimation. In: *CVPR* (2019)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: *CVPR* (2017)
30. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016)
31. Serra, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: *ICML* (2018)
32. Sun, J., Wang, Z., Zhang, S., He, X., Zhao, H., Zhang, G., Zhou, X.: OnePose: One-shot object pose estimation without CAD models. In: *CVPR* (2022)
33. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: *ICML* (2020)
34. Tian, L., Oh, C., Cavallaro, A.: Test-time adaptation for 6D pose tracking. *Pattern Recognition* p. 110390 (2024)
35. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI* **13**(04), 376–380 (1991)
36. van de Ven, G.M., Tuytelaars, T., Tolias, A.S.: Three types of incremental learning. *Nature Machine Intelligence* **4**(12), 1185–1197 (2022)
37. Wang, C., Martín-Martín, R., Xu, D., Lv, J., Lu, C., Fei-Fei, L., Savarese, S., Zhu, Y.: 6-PACK: Category-level 6D pose tracker with anchor-based keypoints. In: *ICRA* (2020)
38. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: DenseFusion: 6D object pose estimation by iterative dense fusion. In: *CVPR* (2019)
39. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6D object pose and size estimation. In: *CVPR* (2019)
40. Wang, R., Wang, X., Li, T., Yang, R., Wan, M., Liu, W.: Query6DoF: Learning sparse queries as implicit shape prior for category-level 6DoF pose estimation. In: *ICCV* (2023)
41. Wołczyk, M., Zajac, M., Pascanu, R., Kuciński, Ł., Miłoś, P.: Continual world: A robotic benchmark for continual reinforcement learning. *NeurIPS* (2021)
42. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. *Science and Systems, Robotics* (2018)

43. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint [arXiv:1708.01547](https://arxiv.org/abs/1708.01547) (2017)
44. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: ICML (2017)
45. Zhou, J., Chen, K., Xu, L., Dou, Q., Qin, J.: Deep fusion transformer network with weighted vector-wise keypoints voting for robust 6D object pose estimation. In: ICCV (2023)



Enhancing Quantum Diffusion Models with Pairwise Bell State Entanglement

Shivalee R. K. Shah and Mayank Vatsa^(✉)

Indian Institute of Technology Jodhpur, Jodhpur, Rajasthan, India
{shah.14,mvatsa}@iitj.ac.in

Abstract. This paper introduces a novel quantum diffusion model designed for Noisy Intermediate-Scale Quantum (NISQ) devices. Unlike previous methods, this model efficiently processes higher-dimensional images with complex pixel structures, even on qubit-limited platforms. This is accomplished through a pairwise Bell-state entangling technique, which reduces space complexity. Additionally, parameterized quantum circuits enable the generation of quantum states with minimal parameters, while still delivering high performance. We conduct comprehensive experiments, comparing the proposed model with both classical and quantum techniques using datasets such as MNIST and CIFAR-10. The results show significant improvements in computational efficiency and performance metrics such as FID, SSIM and PSNR. By leveraging quantum entanglement and superposition, this approach advances quantum generative learning. This advancement paves the way for more sophisticated and resource-efficient quantum diffusion algorithms capable of handling complex data on the NISQ devices.

Keywords: Quantum Machine Learning · Diffusion Models · Quantum Entanglement.

1 Introduction

Quantum computing has seen remarkable progress in recent years, opening up new possibilities for solving intricate computational challenges. Specifically, in the field of image generation and machine learning, Quantum Denoising Diffusion Models (QDDMs) are emerging as a promising technology to enhance both the efficiency and effectiveness of these applications. While traditional (non-quantum) diffusion models are quite capable, they often require extensive parameter tuning and can be computationally demanding [1–3].

Diffusion models gradually transform a simple noise distribution into a complex data distribution through a series of iterative steps. This procedure is inherently computationally intensive, especially as the size and complexity of training dataset grow. Quantum diffusion models, however, capitalize on the unique properties of quantum mechanics—namely superposition and entanglement—to circumvent these challenges[4,5]. Quantum entanglement

facilitates the creation of highly correlated states, which can be efficiently manipulated to perform complex transformations, while superposition permits quantum bits (qubits) to occupy multiple states at once, dramatically enlarging the computational space. These quantum characteristics make diffusion models especially powerful for generative tasks involving large datasets and complex, high-dimensional data.

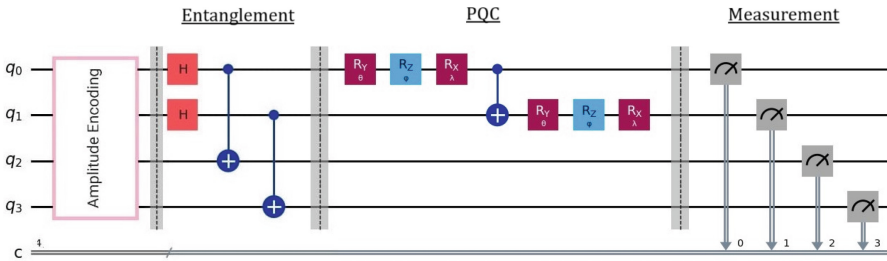


Fig. 1. Schematic overview of the proposed Entanglement Enhanced Quantum Diffusion Model (EEQDM). The circuit illustrates the key components of EEQDM: (1) Amplitude encoding for initial state preparation, (2) Entanglement generation using Hadamard (H) and CNOT gates, enhancing the model’s capability, (3) Parameterized Quantum Circuit (PQC) layer with rotation gates $R_y(\theta)$, $R_z(\phi)$, and $R_x(\lambda)$ and CNOT gate, implementing the diffusion process, and (4) Measurement stage. This architecture leverages pairwise entanglement to enhance the quantum diffusion process, potentially improving performance in tasks such as generative modeling or optimization. Qubits q_0 - q_3 represent the quantum register, while c denotes the classical measurement outcomes.

As shown in Fig. 1, this paper proposes Entanglement Enhanced Quantum Diffusion Model (EEQDM) architecture that leverages the quantum properties through a carefully designed circuit. The circuit begins with amplitude encoding, followed by an entanglement stage using Hadamard (H) gates and CNOT operations. This entanglement facilitates the creation of highly correlated **Bell state pairs**, which can be efficiently manipulated to perform complex transformations. The key feature of the approach is the application of a Parameterized Quantum Circuit (PQC) to only a subset of qubits (q_0 and q_1 in the image). This selective application of the PQC significantly reduces the parameter count while maintaining the power of quantum processing. The circuit concludes with a measurement stage, allowing to extract the processed information. The proposed method is evaluated against traditional classical models and current quantum models using widely recognized datasets like MNIST digits and CIFAR-10, which are frequently utilized in quantum machine learning research. The findings show an increase in computational efficiency and performance metrics, highlighting the effectiveness of the entanglement strategy employed in the proposed model to enhance QDDMs.

2 Literature Review and Related Work

Recent advancements in quantum machine learning have highlighted the potential of Quantum Diffusion Models (QDMs) in improving image generation tasks. Classical diffusion models, such as Denoising Diffusion Probabilistic Models (DDPMs), have been instrumental in advancing image synthesis but are often hampered by high computational demands and the need for extensive parameter tuning [1]. The transition to quantum-based models offers a promising solution to these challenges.

2.1 Classical Diffusion Models

Classical diffusion models are generative models designed to learn the probability distribution $p(x)$ of a dataset, enabling the generation of new samples from this distribution. The diffusion process involves a Markov chain that gradually maps an arbitrary distribution $q(x_0)$ to a simpler, treatable distribution $\pi(x_T)$, often a Gaussian distribution. This is achieved through a forward process using a Markov kernel $q(x_t|x_{t-1})$, and a parametric model is trained to estimate the inverse Markov chain, $p_\theta(x_{t-1}|x_t)$ [1].

2.2 Parameterized Quantum Circuits

Parameterized Quantum Circuits (PQCs) are essential components in the domain of quantum machine learning[24], serving as the quantum analog of classical artificial neural networks. PQCs are composed of quantum gates that perform parametric transformations on quantum states, organized in layers to facilitate complex data processing. Each quantum gate within these circuits is defined by rotation angles, which are trainable parameters optimized using techniques such as gradient descent [6]. The strong entangling ansatz, which combines trainable rotation gates (R_x, R_y, R_z) with C-NOT gates to create entanglement between qubits, has been particularly effective [7].

Training PQCs on current noisy intermediate-scale quantum (NISQ) [22] hardware poses challenges due to high noise levels. Consequently, simulations using software libraries like PennyLane are often employed for training, where both forward computations and optimizations are performed on classical computers. These simulations encode classical data into quantum states using amplitude encoding, which maps a classical vector's components onto the coefficients of a quantum state, allowing the representation of 2^N classical features with N qubits [8]. However, this encoding requires a number of C-NOT gates that grows exponentially with the number of qubits, presenting a scalability challenge.

2.3 Quantum Diffusion Models

Quantum Diffusion Models (QDMs) combine the principles of quantum computing with the methodology of diffusion models to enhance generative capabilities. These models leverage quantum mechanics properties such as superposition

and entanglement to process information more efficiently than classical models. Quantum Denoising Diffusion Models (QDDMs) utilize parameterized quantum circuits to model the data distribution, providing a compact representation that reduces computational load compared to classical counterparts [6].

A significant development in QDDMs is the use of intermediate measurements and ancillary qubits, which have been shown to improve the quality of generated samples by introducing non-linear mappings over state amplitudes [6]. However, excessive measurements can lead to model collapse due to the loss of initial noise information. Hybrid models, combining classical autoencoders with QDDMs, have also demonstrated enhanced performance by simplifying PQCs and enabling implementation on real quantum hardware [8]. These latent models reduce the dimensions of the input data before processing with quantum circuits, improving efficiency but adding complexity by requiring classical pre-processing.

The proposed research further enhances this approach by reducing the parameter count directly within the quantum circuit itself, eliminating the need for classical autoencoders. The Bell state entanglement strategy helps in maintaining the fidelity of quantum states across iterations, reducing errors, and enabling the model to handle datasets with fewer parameters. By benchmarking EEQDM against classical and existing quantum models using datasets like MNIST digits and CIFAR-10, we demonstrate significant improvements in computational efficiency and performance metrics, showcasing the potential of our approach in enhancing QDDMs.

3 Methodology

In this section, we provide the details of the methodology employed to construct EEQDM. The proposed approach integrates the design of a quantum variational circuit with the implementation of a diffusion process enhanced by an entanglement-based technique.

3.1 Construction of Quantum Diffusion Models

The Entanglement-Enhanced Quantum Diffusion Model (EEQDM) introduces a novel quantum circuit design that harnesses the power of Bell-state entanglement to enhance performance while simultaneously reducing the parameter count in quantum diffusion models. The architecture of EEQDM consists of three primary stages: Amplitude Encoding, Pair-wise Bell-state preparation, and Parameterized Quantum Circuit (PQC), followed by a measurement stage.

Input: Amplitude Encoding We begin by performing amplitude encoding to embed the input data into the quantum circuit. This method uses $\log(n)$ qubits, where, n is the number of features in the dataset. Amplitude encoding is efficient for handling high-dimensional data as it maps the classical data vector components onto the amplitude of quantum states. let $|x\rangle$ be the quantum state

representing the input data vector x . Mathematically, amplitude encoding can be expressed as:

$$|x\rangle = \sum_{i=0}^{n-1} x_i |i\rangle \quad (1)$$

We first flatten the 2D image data into a 1D vector, normalize it, and then apply amplitude encoding. This process allows us to represent an image of N pixels using only $(\lceil \log_2(N) \rceil)$ qubits [14, 19], significantly reducing the quantum resources required compared to direct qubit encoding methods. For instance, a 16×16 image (256 pixels) would require only 8 qubits, while a 64×64 image (4096 pixels) would need 12 qubits. This logarithmic scaling in qubit requirements demonstrates the efficiency of amplitude encoding for handling various image sizes in EEQDM.

Entanglement Following the amplitude encoding, the Pair-wise Bell-state preparation stage implements a specific entanglement strategy that creates a unique quantum state, efficiently distributing information across the qubits. This process can be described as "pairwise qubit entanglement" and unfolds as follows:

1. Hadamard gates (H) are applied to the first half of the qubits (excluding the ancilla qubit). This creates an equal superposition state for these qubits, preparing them for entanglement.
2. CNOT gates are then used to entangle each qubit from the first half with a corresponding qubit in the second half. Specifically, for n qubits (excluding the ancilla), we apply CNOT gates where, the control qubits are from the first half (indices 0 to $n/2-1$) and the target qubits are their corresponding partners in the second half (indices $n/2$ to $n-1$).

The Pair-wise Bell-state preparation is crucial to EEQDM's parameter reduction mechanism. By creating specific entanglement pairs between the first and second half of the qubit register, we establish information pathways that allow the subsequent Parameterized Quantum Circuit (PQC) to operate on a reduced set of qubits while still accessing information from the entire input state. Thus, significantly reducing the number of parameters required while maintaining the model's expressive power.

The effectiveness of this approach is evident in the parameter reduction graph Fig. 2, which shows a consistent 40-47% reduction in parameters for qubit counts between 8 and 18, compared to existing quantum diffusion models. This substantial reduction in parameters, enabled by the unique entanglement strategy, is a key factor in EEQDM's improved efficiency and scalability.

Ansatz: Parameterized Quantum Circuit The foundational element of EEQDM architecture is a Parameterized Quantum Circuit (PQC) which serves as the ansatz. We conduct an extensive exploration of various circuit depths and

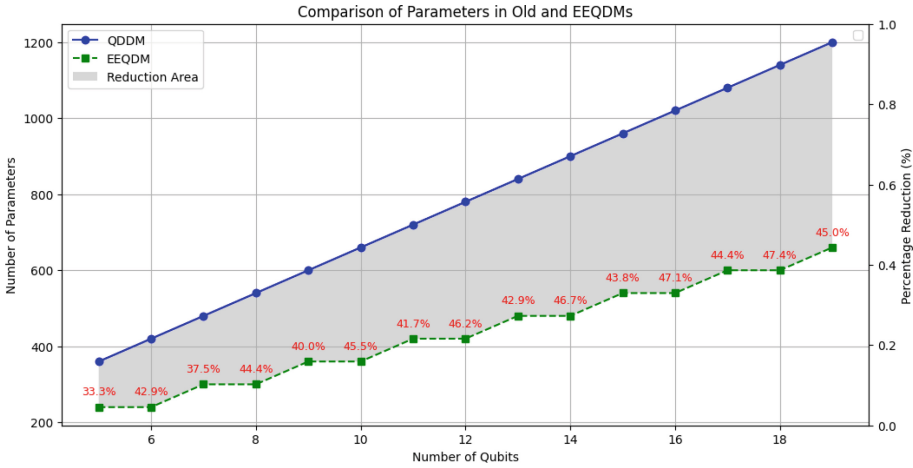


Fig. 2. Comparison of parameters in QDDM[6] and the proposed EEQDM as the number of qubits increases. The shaded area highlights the parameter reduction achieved by the new model.

entanglement architectures within these PQCs to ascertain the optimal configuration for the proposed quantum diffusion model. Each layer of the ansatz is composed of trainable parameters, facilitated through rotation and C-NOT gates, enabling the circuit to dynamically adapt and learn from the data throughout the training process [15]. In our experiments, we vary the depth (L) and entanglement patterns within the PQCs to determine the configuration that best aligns with the requirements of the diffusion model [16]. This framework permits the circuit to effectively adapt to the data distribution across different noise levels encountered during the diffusion process. The selection of this ansatz is strategically motivated by the need to balance expressivity with parameter efficiency, which is essential for streamlined training and to prevent overfitting [17]. This balance is crucial for optimizing the learning capabilities of our model while ensuring generalization across diverse datasets [18].

3.2 Diffusion Process

The diffusion process follows a Markov chain framework, where, data undergoes forward diffusion to introduce noise, followed by reverse diffusion to denoise and reconstruct the data. This process is inspired by classical diffusion models but adapted to the quantum domain.

Forward Diffusion Noise is incrementally added to the data across multiple timesteps. The forward diffusion process can be mathematically represented as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I) \tag{2}$$

where, α_t controls the variance schedule.

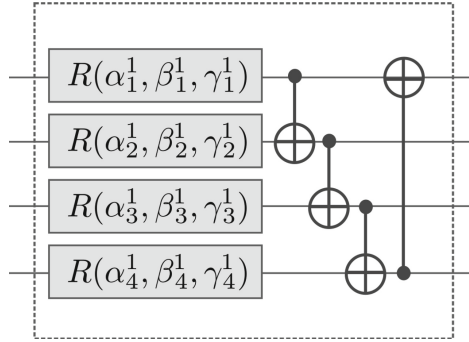


Fig. 3. Example of ansatz with 4-qubit strongly entangling layers ($L = 1$) showing rotations R and CNOTs. In practice, we use multiple layers and optimize the depth for our specific task. Image from PennyLane documentation, available at: [PennyLane Documentation](#).

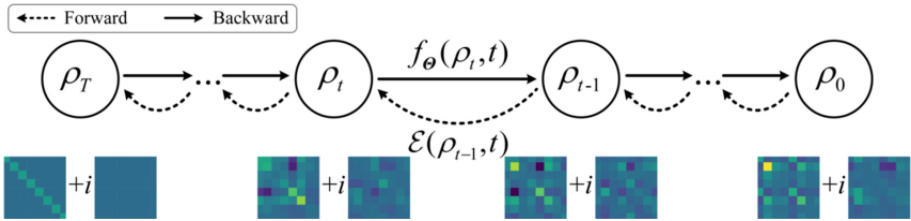


Fig. 4. It depicts the state evolution from ρ_T to ρ_0 through intermediate states ρ_t and ρ_{t-1} . Each step involves applying the function $\mathcal{E}(\rho_{t-1}, t)$ and the quantum operation $f_\theta(\rho, t)$. The bottom row showcases visual representations of these states, highlighting the transformation and diffusion process in the quantum generative model.[23]

Reverse Diffusion The reverse diffusion process, also known as the sampling or generation process, is a crucial component of diffusion models. This process involves gradually denoising a random input to produce a high-quality sample. Our implementation of reverse diffusion is inspired by recent advancements in the field [1, 6, 7].

The process begins with a random noise tensor x_0 , typically sampled from a standard normal distribution. This noise is then iteratively refined through T steps, where, T is the number of diffusion steps:

$$x_0 \sim \mathcal{N}(0, I) \tag{3}$$

$$x_1, x_2, \dots, x_T = \text{ReverseProcess}(x_0) \tag{4}$$

At each step t , the proposed model predicts either the denoised data directly or the noise component, depending on the chosen prediction goal. For data prediction, we directly use the network output:

$$x_t = f_\theta(x_{t-1}) \quad (5)$$

where, f_θ is our quantum circuit with parameters θ . This direct prediction of denoised data, rather than noise, has empirically shown better results in experiments. Our implementation also supports conditional generation, where, additional label information can be provided to guide the reverse diffusion process. This is particularly useful for tasks requiring controlled generation or class-specific sampling.

3.3 Training and Optimization

Training involves optimizing the PQC parameters to minimize the reconstruction error. We use the Mean Squared Error (MSE) loss function to quantify the difference between the reconstructed and original data:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|^2 \quad (6)$$

where, x_i and \hat{x}_i represent the original and reconstructed data, respectively.

Our hybrid quantum-classical pipeline integrates quantum computing capabilities with classical optimization techniques to solve complex problems efficiently. At its core, a parameterized quantum circuit processes and encodes input data into quantum states. The circuit's output is measured and classically post-processed to compute the objective function. The circuit parameters are optimized using the classical Adam optimizer. We conducted hyperparameter tuning for the learning rate, testing values of 0.1, 0.01, and 0.001, with 0.1 yielding the best results. This iterative process of quantum computation followed by classical optimization allows us to harness the potential quantum advantage in data processing while leveraging well-established classical algorithms for parameter updates. This hybrid approach enables us to tackle problems that may be challenging for purely classical or quantum methods, potentially opening new avenues in optimization and machine learning tasks.

Performance metrics such as Structural Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR) are employed to evaluate model performance. These metrics assess the model's ability to generate high-quality images that closely resemble the original data distribution.

4 Experimental Setup

4.1 Datasets

In our experiments, we utilized two standard benchmark datasets widely recognized in both classical and quantum machine learning communities: MNIST [11] and CIFAR-10[12]. These datasets were chosen for their widespread use in evaluating image processing models, including recent quantum machine learning

approaches [10, 13, 20]. The MNIST dataset consists of 70,000 grayscale images of handwritten digits (28×28 pixels), split into 60,000 training images and 10,000 test images. This dataset has been extensively used in quantum machine learning literature due to its simplicity and the clear benchmarking it provides for digit recognition tasks [10, 20]. CIFAR-10 includes 60,000 color images (32×32 pixels) across ten different classes, with 50,000 training images and 10,000 test images. Its inclusion allows us to evaluate our model's performance on more complex, real-world images, following recent trends in quantum image processing research [13, 20].

To assess EEQDM's ability to handle varying data dimensionality and to ensure compatibility with different qubit counts in our quantum system, we pre-processed the images to three different resolutions: 8×8 , 16×16 , and 32×32 pixels. This approach not only aligns with recent quantum image processing studies [10, 13] but also challenges the model to manage increasingly larger feature sets effectively, providing insights into its scalability and performance across different data complexities.

4.2 Metrics

Loss Curves We analyze the training and validation loss curves to evaluate the model's learning progress and generalization ability. The loss function used is Mean Squared Error (MSE), defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (7)$$

where, y_i are the true values and \hat{y}_i are the predicted values.

Structural Similarity Index (SSIM) [21] SSIM assesses the perceived quality of generated images compared to the originals:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (8)$$

where, μ_x and μ_y are the average pixel intensities, σ_x and σ_y are the standard deviations, and σ_{xy} is the covariance of pixels in images x and y . SSIM ranges from -1 to 1, with 1 indicating perfect structural similarity.

Peak Signal-to-Noise Ratio (PSNR) PSNR quantifies the ratio between the maximum possible signal power and the power of distorting noise:

$$PSNR = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (9)$$

where, MAX_I is the maximum possible pixel value, and MSE is the mean squared error between the generated and original images. Higher PSNR values

generally indicate better reconstruction quality. This comparison helps us understand the learning behavior in both quantum and classical domains, as well as the specific benefits and trade-offs of the proposed quantum diffusion approach.

Fréchet Inception Distance (FID) [9] FID measures the similarity between the distribution of generated images and that of real images. It is computed as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

where μ_r and μ_g are the mean feature representations, and Σ_r and Σ_g are the covariance matrices for real and generated images, respectively. Lower FID scores indicate higher quality and diversity.

5 Results

5.1 Comparison of EEQDM and QDDM

Model Specifications : The proposed Entanglement-Enhanced Quantum Diffusion Model (EEQDM) incorporates a novel pairwise qubit entanglement technique aimed at improving scalability and efficiency. We compared EEQDM with an existing Quantum Denoising Diffusion Model (QDDM) that does not include this feature. Both models were evaluated using 8×8 and 16×16 MNIST images, as well as 16×16 CIFAR-10 images, across depths ranging from 10 to 50.

Evaluation of Models : As depicted in Fig. 5 and Fig. 6, a detailed comparison of performance metrics and computational efficiency reveals several key trends:

- For 8×8 MNIST images, the performance difference between EEQDM and QDDM is minimal, with QDDM slightly outperforming EEQDM at higher depths in terms of loss values. Both models have comparable execution times, with EEQDM being marginally faster. This indicates that for simpler, smaller-scale tasks, both models perform adequately without significant distinctions.
- When processing larger and more complex images, the advantages of EEQDM become evident. For 16×16 MNIST images, EEQDM consistently outperforms QDDM across all metrics, including lower final loss, higher SSIM, and improved PSNR values. This superior performance is achieved with notably faster execution times, especially as model depth increases, suggesting that EEQDM’s entanglement feature not only enhances image reconstruction quality but also improves computational efficiency for higher-resolution grayscale images.
- The most significant contrast is observed with 16×16 CIFAR-10 color images. EEQDM demonstrates clear superiority in both performance metrics and computational efficiency. It achieves better image reconstruction quality (evidenced by improved loss, SSIM, and PSNR values) while requiring significantly less execution time compared to QDDM. The computational advantage is particularly pronounced, with EEQDM processing these complex images

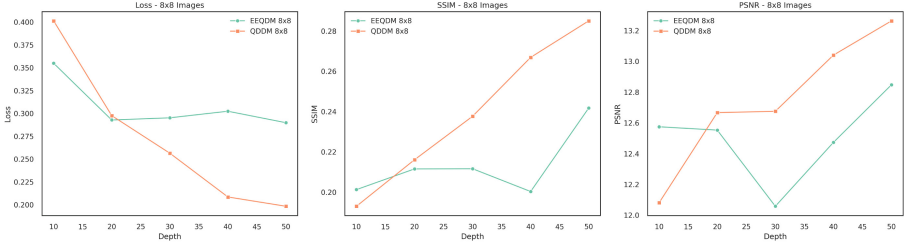
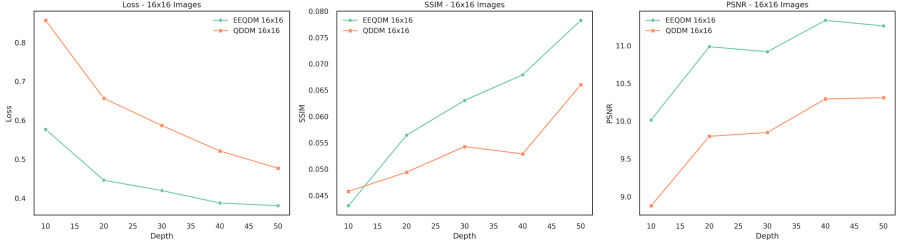
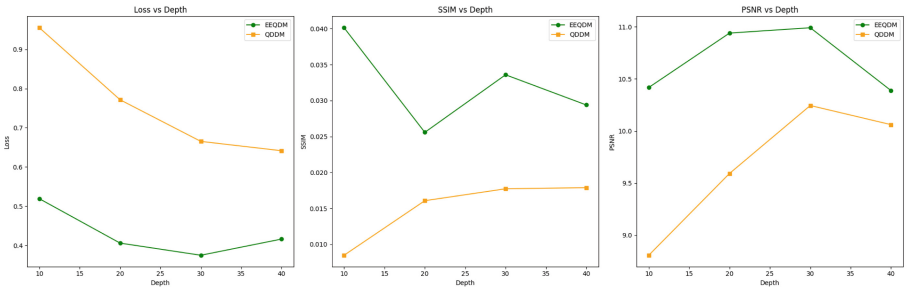
(a) Performance comparison for 8×8 MNIST images.(b) Performance comparison for 16×16 MNIST images.(c) Performance comparison for 16×16 CIFAR10 images.

Fig. 5. Performance comparison of EEQDM and QDDM for 8×8 and 16×16 images across depths from 10 to 50. Results indicate EEQDM's overall superior performance, particularly at greater depths and complex data.

nearly twice as fast as QDDM at higher depths. This suggests that the pairwise entanglement feature effectively manages the increased computational demands of more complex tasks, offering a better balance between reconstruction quality and processing time.

5.2 Comparison with Classical Denoising Diffusion Model

Model Specifications : The classical Denoising Diffusion Probabilistic Model (DDPM) is a generative model that incrementally denoises data to produce

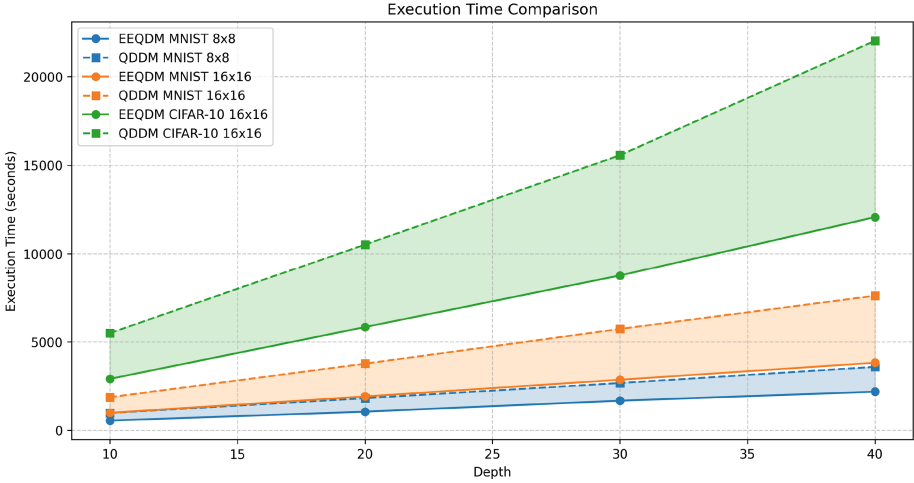


Fig. 6. Execution Time Comparison of EEQDM and QDDM for 8×8 and 16×16 Images Across Various Depths demonstrating that EEQDM consistently outperforms QDDM in terms of execution speed across all depths and configurations, with the performance gap widening as the network depth and data complexity increases.

images. It employs a U-Net architecture with an encoder-decoder design, featuring convolutional layers with ReLU activations. The model includes three depth variants with 2 or 3 layers and different initial channel counts. Designed for 16×16 pixel images, the DDPM variants contain between 892 and 6,781 trainable parameters. This classical model serves as a baseline for comparison with our quantum-inspired EEQDM approach, aligning with recent work in quantum-classical comparisons for image generation [6].

Evaluation of Models : Table 1 provides a comparison between EEQDM and the classical models. The results reveal intriguing performance trade-offs:

- EEQDM demonstrates superior loss reduction as the number of parameters increases, ultimately achieving lower loss with fewer parameters than its classical counterparts. However, this improved performance comes at a significant cost in terms of computation time. EEQDM’s execution time increases exponentially with the parameter count, while classical models maintain relatively constant and much lower execution times regardless of parameter increases.
- Despite the classical models’ speed advantage, they show limited improvement in loss reduction even as their complexity grows. This creates a clear trade-off between model accuracy and computational efficiency. EEQDM offers potentially higher accuracy but requires substantially more processing time, particularly for larger parameter sets.

Table 1. Comparison of EEQDM and Classical Models.

Model Type	Depth/Config	Num Params	Final Loss	Execution Time (s)
EEQDM	10	150	0.3674	994.36
EEQDM	20	300	0.2739	1934.08
EEQDM	30	450	0.2697	2925.98
EEQDM	40	600	0.2560	3991.76
EEQDM	50	750	0.2328	5096.88
Classical	depth2	892	1.0154	175.99
Classical	depth3_channels2	1735	1.0058	230.34
Classical	depth3_channels4	6781	1.0033	252.70

5.3 Comparison with Previous Models on CIFAFR10

To evaluate the performance of our proposed EEQDM model, we compared it with several state-of-the-art models from previous studies, including U-Net, QU-Net, and Q-Dense. EEQDM exhibits superior performance across all measured metrics while utilizing fewer parameters.

- With 750 parameters compared to QDDM’s 1350, EEQDM achieves a 34% reduction in final loss (0.2993 vs. 0.4536), a 157% improvement in Structural Similarity Index (SSIM) (0.0433 vs. 0.0169), and a 0.48 dB increase in Peak Signal-to-Noise Ratio (PSNR) (10.65 dB vs. 10.17 dB). Moreover, EEQDM’s execution time is reduced by 35.9% (5932 seconds vs. 9248 seconds), indicating substantial computational efficiency gains.
- The classical model, despite utilizing 6781 parameters, achieves an average loss of 0.9763, whereas EEQDM, with only 750 parameters, attains a final loss of 0.2993—a 69.3% reduction. The disparity in image quality metrics is even more pronounced: EEQDM’s PSNR of 10.65 dB significantly outperforms the classical model’s -46.77 dB.

5.4 FID Score Comparison of EEQDM and QDDM on MNIST Dataset

The mean FID score for the EEQDM model is 382.36 with a standard deviation of 74.66, indicating moderate variability around the mean. In comparison, the mean FID score for the QDDM model is 420.46 with a lower standard deviation of 44.10, suggesting that the scores are more tightly clustered around the average (see Table 2 for class-wise comparison).

Table 2. FID Score Comparison for Each Digit.

Digit	FID_ EEQDM	FID_ QDDM
0	330.93	459.59
1	373.53	402.79
2	401.67	466.97
3	321.32	388.38
4	550.58	406.61
5	443.47	476.37
6	319.73	381.72
7	327.35	433.68
8	331.05	449.49
9	423.96	339.02

6 Conclusion

This paper introduces the Entanglement-Enhanced Quantum Diffusion Model (EEQDM). It offers clear advantages in processing complex, high-resolution data. EEQDM delivers superior FID, SSIM and PSNR metrics, reduced parameter counts, and faster execution times compared to existing quantum models. Its innovative use of pairwise entanglement is highly effective for intricate data structures, making it more resource-efficient in the quantum domain. An important direction for future work is to develop encoding methods that preserve spatial correlations, enabling correlated data points to be placed within the same entanglement pairs. This approach could enhance model efficiency by leveraging inherent data structures, optimizing quantum resources, and maintaining coherence in entangled states. Moreover, further optimization of quantum circuits is needed to match the speed of classical models. As quantum hardware advances, EEQDM's potential to revolutionize quantum machine learning and complex data processing will grow. Future research should explore EEQDM's applications in other quantum machine learning tasks.

References

1. Dhariwal, P., & Nichol, A. (2021). Diffusion Models Beat GANs on Image Synthesis. arXiv preprint [arXiv:2105.05233](https://arxiv.org/abs/2105.05233)
2. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. **1**, 4 (2021)
3. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. *Adv. Neural. Inf. Process. Syst.* **33**, 6840–6851 (2020)
4. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. *Nature* **549**(7671), 195–202 (2017)
5. Gabor, T., Sunkel, L., Ritz, F., Phan, T., Belzner, L., Roch, C., Feld, S., & Linnhoff-Popien, C. (2020). The holy grail of quantum artificial intelligence: Major challenges in accelerating the machine learning pipeline

6. Koelle, M., Stenzel, G., Stein, J., Zielinski, S., Ommer, B., & Linnhoff-Popien, C. (2024). Quantum Denoising Diffusion Models. arXiv preprint [arXiv:2401.07049](https://arxiv.org/abs/2401.07049)
7. Kim, D., & Kang, S. (2023). Quantum Denoising Diffusion Probabilistic Models for Image Generation. Korean Conference on Semiconductors
8. Koelle, M., Stenzel, G., Stein, J., Zielinski, S., Ommer, B., & Linnhoff-Popien, C. (2023). Enhancing Quantum Diffusion Models with Intermediate Measurements. arXiv preprint [arXiv:2310.05866](https://arxiv.org/abs/2310.05866)
9. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Adv. Neural. Inf. Process. Syst.* **30**, 6626–6637 (2017)
10. Adhikary, S., Dangwal, S., & Bhowmik, D. (2024). Supervised learning on qubits with natural gradient descent and quantum geometric tensor. arXiv preprint [arXiv:2401.07049](https://arxiv.org/abs/2401.07049)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
12. Krizhevsky, A. (2009). The CIFAR-10 dataset. Retrieved from <https://www.cs.toronto.edu/~kriz/cifar.html>
13. Cao, S., et al. (2023). Quantum generative adversarial networks for image generation: A survey. arXiv preprint [arXiv:2310.05866](https://arxiv.org/abs/2310.05866)
14. Möttönen, M., Vartiainen, J. J., Bergholm, V., & Salomaa, M. M. (2004). Transformation of quantum states using uniformly controlled rotations
15. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. *Quantum Science and Technology* **4**(4), 043001 (2019)
16. M. Cerezo et al., "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp
17. Sim, S., Johnson, P.D., Aspuru-Guzik, A.: Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies* **2**(12), 1900070 (2019)
18. Kandala, A., et al.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**(7671), 242–246 (2017)
19. Steane, A.: Quantum computing. *Rep. Prog. Phys.* **61**(2), 117–173 (1998)
20. Feng, Y., Sachdev, S., Kalsi, S., Chen, H., & Rebertrost, P. (2023). Quantum versus tensor network algorithms for machine learning on the mnist dataset. arXiv preprint [arXiv:2311.15444](https://arxiv.org/abs/2311.15444)
21. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
22. Preskill, J.: Quantum Computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
23. Chen, C., & Zhao, Q. (2024). Quantum Generative Diffusion Model. arXiv preprint [arXiv:2401.07039](https://arxiv.org/abs/2401.07039)
24. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. *Contemp. Phys.* **56**(2), 172–185 (2015)



Efficient Prescribed-Time and Robust Zeroing Neural Networks for Computing Time-Variant Plural Stein Matrix Equation

ShuPeng Li^{1,2}(✉) and ZhaoHui Qi^{1,2}

¹ College of Information Science and Engineering Academy, Hunan Normal University, Changsha 410081, China
937847754@qq.com

² Institute of Interdisciplinary Studies, Hunan Normal University, Changsha 410081, China
<https://www.hunnu.edu.cn/>

Abstract. This paper presents two novel models, designated as efficient prescribed-time and robust zeroing neural network (EPTR-ZNN), which employ a novel activation function (AF) and adaptive dynamic parameter (ADP) to address the time-variant plural Stein matrix equation (TV-PSME). The EPTR-ZNN model is proposed by combining the novel AF with the standardized ZNN design process. In comparison to traditional fixed parameter (FP) ZNN models, the EPTR-ZNN model exhibits a faster convergence rate, enhanced computational efficiency, and stronger robustness. To further improve these performance characteristics, we replaced the FP in EPTR-ZNN model with an ADP to develop the EPTR-DPZNN model. In contrast to traditional divergent dynamic parameters (DPs), the ADP can be adjusted in a synchronous manner as the model converges, thereby enhancing the computational efficiency. Theoretical analysis verifies the prescribed-time convergence and robustness of the EPTR-ZNN and EPTR-DPZNN models. Finally, simulation experiments demonstrate that the EPTR-ZNN model exhibits accelerated convergence compared to other ZNN models, while the EPTR-DPZNN model exhibits the best convergence performance, higher computational efficiency, and stronger robustness to time-variant bounded noise (TV-BN) and time-variant unbounded noise (TV-UN).

Keywords: Recurrent neural network · Adaptive dynamic parameter · Time-variant plural Stein matrix equation · Prescribed-time convergence · Robustness

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_24.

1 Introduction

The Stein equation represents a significant mathematical equation in the field of dynamic control systems. It is employed extensively in various domains of science and engineering, including optimization [1], control system design [2] and robot motion planning [3]. There are two approaches to solving the Stein equation: numerical algorithms (NAs) and recurrent neural networks (RNNs). NAs, being inherently sequential, are inadequate for time-variant and high-dimensional problems [4], making them unsuitable for real-time solutions of TV-PSME. On the other hand, RNNs, with their online computation and parallel processing capabilities, are hardware-implementable and have been widely used to tackle complex computational issues [5]. Specifically, gradient-based RNNs (GRNNs) have been extensively used for time-invariant challenges [5]. However, GRNNs fail to solve TV-PSME effectively due to their inability to adapt to the evolving coefficients at the required pace, resulting in persistent and non-declining residuals over time.

To address these challenges, Zhang *et al.* proposed the zeroing neural network (ZNN) as a novel general RNN for addressing time-variant problems [6]. In light of his seminal work, numerous enhanced and efficacious ZNN models have been developed over the past two decades. In paper [6], the ZNN model converges exponentially. To accelerate this rate, Li *et al.* [7] proposed a finite-time convergence ZNN (FNTC-ZNN) model. However, while the convergence time of FNTC-ZNN is finite, there is no definitive upper limit and its convergence time varies with the initial state of the model. Xiao *et al.* [8] developed a prescribed-time convergence ZNN model. Its convergence time is solely dependent on the parameters of the model and is independent of the initial state. Moreover, its convergence time can be guaranteed to always converge before the prescribed time constant.

In addition to convergence time, the design of convergence parameters (CPs) and robustness are also important criteria for evaluating the performance of ZNN models. Original ZNN (O-ZNN) models effectively address the TV-PSME but overlook the prevalent external interference in real-world settings. Dai *et al.* [9] introduced a PTR-ZNN model to address convergence in noise, yet it struggles with specific noise types. The PTR-ZNN model's fixed parameters (FP) are impractical for dynamic hardware systems. Xiao *et al.* [10] proposed an IENT-ZNN model with a DP, enhancing convergence and robustness while mitigating the FP issue. However, the inclusion of integral terms complicates the model's structure. Furthermore, the DPs in many conventional DP-ZNN models are divergent [10, 11], which leads to a waste of computational resources. Although the aforementioned ZNN models demonstrate enhanced convergence rates and robustness, they concomitantly entail a proportional increase in computational cost. To address this issue, Lou *et al.* [12] proposed a convergent HTPR-ZNN model. However, this model demonstrated substandard convergence performance and robustness. To enhance computational efficiency while maintaining convergence and robustness, this paper proposes two ZNN models based on a novel AF and ADP: the efficient prescribed-time robust ZNN (EPTR-

ZNN) model and the EPTR-ZNN model with an ADP (EPTR-DPZNN). The novel AF accelerates the convergence rate, and the ADP enhances computational efficiency and robustness. Consequently, the EPTR-ZNN model exhibits accelerated prescribed-time convergence compared to traditional ZNN models, while the EPTR-DPZNN model demonstrates a faster convergence rate, enhanced computational efficiency, and stronger robustness to TV-BN and TV-UN. Table 1 shows the comparison between different ZNN models using different CPs.

Table 1. Comparison between ZNN models.

Model	PTR-ZNN [9]	PCCV-ZNN [11]	HTVPR-ZNN [12]	EPTR-ZNN	EPTR-DPZNN
Value range	Real-number	Real-number	Real-number	Plural	Plural
Type of CP	Fixed	Divergent	Convergent	Fixed	Adaptive
Convergence	Prescribed-time	Finite-time	Prescribed-time	Accelerated prescribed-time	Better Prescribed-time
Computational efficiency	Slow	Moderate	Slowest	High	Higher
TV-BN robustness	Moderate	Weak	Weak	Moderate	Strong
TV-UN robustness	Weak	Weaker	Weaker	Moderate	Strong

The structure of this paper is organized as follows: Section 2 introduces the problem formulation to be solved. Section 3 details the construction of the two efficient prescribed-time and robust ZNN models. Section 4 demonstrates the global stability, prescribed-time convergence and robustness of the EPTR-ZNN and EPTR-DPZNN models. Section 5 conducts simulation experiments to prove the above theory. The paper is concluded in Section 6. The main contributions of this paper are as follows.

1. A novel activation function and adaptive dynamic parameter is proposed. On this basis, A fast convergence EPTR-ZNN model and a robust EPTR-DPZNN model with high computational efficiency are designed to compute the TV-PSME.
2. Through theoretical derivation, the global stability, prescribed-time convergence and robustness of the EPTR-ZNN and EPTR-DPZNN models are demonstrated. Furthermore, the upper bound for the prescribed convergence time is also calculated.
3. Simulation experiments indicate that the EPTR-ZNN and EPTR-DPZNN models have faster convergence than traditional models, while the EPTR-DPZNN model also exhibits higher computational efficiency and stronger robustness.

2 Problem statement

This section introduces the description of the time-variant plural Stein matrix equation (TV-PSME) problem. The TV-PSME with time-variant coefficients is generally defined as follows:

$$O(t)X(t)P(t) + X(t) = Q(t), \quad (1)$$

where $O(t), P(t)$ and $Q(t) \in \mathbb{C}^{k \times k}$ are known full rank plural matrices, and $X(t) \in \mathbb{C}^{k \times k}$ is an unknown plural matrix awaiting solution. If $O(t) = P^T(t)$, TV-PSME (1) can be converted to the time-variant Lyapunov matrix equation, and it can also be transformed into the time-variant Sylvester matrix equation, if $P(t)$ is a nonsingular matrix [9].

3 Efficient prescribed-time and robust ZNN models

3.1 EPTR-ZNN model

According to the standard ZNN design process [6], the EPTR-ZNN model is constructed in three steps.

First, define an error function based on the equation (1) to be solved.

$$E(t) = O(t)X(t)P(t) + X(t) - Q(t) = E_r(t) + iE_m(t). \tag{2}$$

Here, $E(t) \in \mathbb{C}^{k \times k}$ and its real and imaginary parts are $E_r(t)$ and $E_m(t)$. Next, in order for $E(t)$ to converge to zero, the evolution formula is described as

$$\dot{E}(t) = -\lambda \left(\Phi(E_r(t)) + i\Phi(E_m(t)) \right), \tag{3}$$

where $\dot{(\cdot)}$ denotes the time derivative and convergence parameter (CP) $\lambda > 0$ is used to control the convergence rate of ZNN models. $\Phi(\cdot) \in \mathbb{C}^{k \times k} \rightarrow \mathbb{C}^{k \times k}$ is a mapping of a matrix-valued AF, whose elements are denoted by $\phi(\cdot) \in \mathbb{C} \rightarrow \mathbb{C}$. The AF plays a crucial role in the convergence and robustness of ZNNs. The AF must be a monotonically increasing odd function. The larger the slope of the AF, the faster the model converges. Therefore, the function of AF is to increase its slope as much as possible within the acceptable calculating pressure range. To improve the convergence performance and robustness of the EPTR-ZNN model, this paper designs a novel fast prescribed (FP) AF as follows:

$$\phi_{fp}(y) = \left(\frac{h_1}{p} |y|^{(1-p)} \exp(|y|^p) + h_2 |y|^q \exp(|y|^{q-1}) \right) \text{sgn}(y) + h_3 y^r + h_4 \text{sgn}(y), \tag{4}$$

where $h_1 > 0, h_2 > 0, h_3 \geq 0, h_4 \geq 0, q > 1, 0 < p < 1, r$ is a positive odd number, they are used to adjust the convergence rate and noise resistance of the AF and the function $\text{sgn}(y)$ is as shown.

$$\text{sgn}(y) = \begin{cases} 1, & y > 0, \\ 0, & y = 0, \\ -1, & y < 0. \end{cases} \tag{5}$$

In the FP AF, $\left(h_1 |y|^{(1-p)} \exp(|y|^p) / p + h_2 |y|^q \exp(|y|^{q-1}) \right) \text{sgn}(y)$ is the convergent term, where $h_1 |y|^{(1-p)} \exp(|y|^p) \text{sgn}(y) / p$ plays the main convergence role when $y < 1$, $h_2 |y|^q \exp(|y|^{q-1}) \text{sgn}(y)$ plays the main convergence role when

$y > 1$, and $h_3y^r + h_4\text{sgn}(y)$ is the noise resistance term, where h_3y^r and $h_4\text{sgn}(y)$ are used to resist TV-UN and TV-BN respectively. If in a noiseless environment, $h_3 = h_4 = 0$. The FP AF is expressed as:

$$\phi_{fp1}(y) = \left(\frac{h_1}{p} |y|^{(1-p)} \exp(|y|^p) + h_2 |y|^q \exp(|y|^{q-1}) \right) \text{sgn}(y) \tag{6}$$

Last, by substituting error function (2) and the FP AF (4) into evolution formula (3), the EPTR-ZNN model can be obtained:

$$\begin{aligned} O(t)\dot{X}(t)P(t) + \dot{X}(t) &= -\lambda \left(\Phi_{FP}(E_r(t)) + i\Phi_{FP}(E_m(t)) \right) \\ &\quad - \dot{O}(t)X(t)P(t) - O(t)X(t)\dot{P}(t) + \dot{Q}(t). \end{aligned} \tag{7}$$

3.2 EPTR-DPZNN model

In order to evaluate the performance of the ZNN model, this paper employs the ode45 solver in Matlab. In order to enhance the computational efficiency of the EPTR-ZNN model and to improve its convergence rate and robustness, we have introduced the EPTR-DPZNN model. Before delving into the details of the EPTR-DPZNN model, it is essential to have a fundamental understanding of the principles of ode45 solver.

Research indicates that the ode45 is an adaptive step-size solver that dynamically adjusts the step length in response to the change rate in the model’s state [13]. When the state exhibits a smooth variation, the step size is increased to prevent unnecessary computations, thereby improving computational efficiency. When the state experiences significant changes, the step size is reduced to increase the computational workload, thus enhancing the accuracy of the calculations. With regard to the model (7), the change rate of the state matrix $X(t)$ is directly proportional to $\dot{E}(t)$, thus the step size is inversely proportional to $\|\dot{E}(t)\|_F$.

In previous researches, the FP were unable to dynamically adjust the step size in response to changes in the model’s state [9]. While the traditional divergent DP reduced the step size of the ode45 solver, it simultaneously increased the model’s computational cost [11]. To reduce unnecessary computations and thereby enhance computational efficiency, we propose a novel adaptive dynamic parameter (ADP) $\lambda_{adp}(t)$:

$$\lambda_{adp}(t) = \lambda \exp\left(\arctan(\|E(t)\|_F/\alpha)\right), \tag{8}$$

where $\lambda, \alpha > 0$, and $\|E(t)\|_F$ represents the Frobenius norm for the error $E(t)$. The ADP $\lambda_{adp}(t)$ appropriately allocates computational resources by adjusting the value of $\|E(t)\|_F$, thereby enhancing computational efficiency. During the initial phase of convergence, a larger $\|E(t)\|_F$ accelerates the convergence of the model. At the conclusion of the convergence phase, the decreasing $\|E(t)\|_F$

reduces the computational cost of the model. This reasonable allocation of computing resources not only accelerates the convergence but also minimizes unnecessary calculations. By substituting the $\lambda_{adp}(t)$ (8) into Eq. (7), the EPTR-DPZNN model can be obtained.

$$\begin{aligned}
 O(t)\dot{X}(t)P(t) + \dot{X}(t) = & -\lambda_{adp}(t)\left(\Phi_{FP}(E_r(t)) + i\Phi_{FP}(E_m(t))\right) \\
 & - \dot{O}(t)X(t)P(t) - O(t)X(t)\dot{P}(t) + \dot{Q}(t).
 \end{aligned} \tag{9}$$

4 Theoretical analysis

4.1 Global stability analysis

Theorem 1. *Based on the Lyapunov stability theory, the error $E(t)$ of the EPTR-ZNN and EPTR-DPZNN models can be globally converged to zero.*

Proof. Let $e(t)$ denote the element in $E(t)$. The real and imaginary parts are $e_r(t)$ and $e_m(t)$. Let $l(t)$ represent the element of $e(t)$ with the maximum absolute value, i.e., $-|l(t)| \leq e(t) \leq |l(t)|$. The formula (3) is as follows:

$$\begin{cases} \dot{e}_r(t) = -\lambda\phi(e_r(t)), \\ \dot{e}_m(t) = -\lambda\phi(e_m(t)). \end{cases} \tag{10}$$

Define two Lyapunov candidate functions for $\dot{e}_r(t)$ and $\dot{e}_m(t)$.

$$\begin{cases} l_r(t) = \frac{1}{2}e_r^2(t), \\ l_m(t) = \frac{1}{2}e_m^2(t), \end{cases} \text{ and } \begin{cases} \dot{l}_r(t) = -\lambda e_r(t)\phi(e_r(t)), \\ \dot{l}_m(t) = -\lambda e_m(t)\phi(e_m(t)). \end{cases} \tag{11}$$

Clearly, $l_r(t)$ and $l_m(t)$ are positive definite, while $\dot{l}_r(t)$ and $\dot{l}_m(t)$ are negatively definite because $\lambda > 0$ and $\phi(\cdot)$ is a monotonically increasing odd function. According to Lyapunov stability theory [6], $e_r(t)$ and $e_m(t)$ converge to zero. That is, in the sense of Lyapunov, the error of EPTR-ZNN model can converge to zero globally. Similarly, for EPTR-DPZNN model, the same conclusion can be obtained by replacing λ with $\lambda_{adp}(t)$ (8) for Eqs. (10) and (11).

4.2 Prescribed-time convergence analysis

Theorem 2. *In a noiseless environment, the state matrix $X(t)$ of the EPTR-ZNN and EPTR-DPZNN models, initiated from an arbitrary initial state $X(0) \in \mathbb{C}^{k \times k}$, can converge to the theoretical solution within the prescribed-time T_c .*

Proof. Let $\mu = |l(t)| = l(t)\text{sgn}(l(t))$ where $l(t)$ is defined as in Theorem 1. When $\mu = 0$, $e_r(t)$ and $e_m(t)$ also converge to zero. Because there is no external noise, $h_3 = h_4 = 0$. Combining FP AF (6), The derivative of μ is as follows:

$$\begin{aligned}
 \frac{d\mu}{dt} &= \frac{d|l(t)|}{dt} = \frac{d|l(t)|}{dl(t)} \frac{dl(t)}{dt} = \dot{l}(t)\text{sgn}(l(t)) = -\lambda\phi_{fp1}(l(t))\text{sgn}(l(t)) \\
 &= -\lambda\phi_{fp1}(\mu) \Rightarrow \frac{d\mu}{\phi_{fp1}(\mu)} = -\lambda dt.
 \end{aligned} \tag{12}$$

When $t \rightarrow 0, \mu \rightarrow |l(0)|$; when $t \rightarrow T_c, \mu \rightarrow 0$. Solve the above differential equation from $t = 0$ to $t = T_c$:

$$\int_{|l(0)|}^0 \frac{d\mu}{\phi_{fp1}(\mu)} = - \int_0^{T_c} \lambda dt \Rightarrow T_c = \frac{1}{\lambda} \int_0^{|l(0)|} \frac{d\mu}{\phi_{fp1}(\mu)} \leq \frac{1}{\lambda} \int_0^{+\infty} \frac{d\mu}{\phi_{fp1}(\mu)}. \tag{13}$$

Let $\int_0^{+\infty} 1/\phi_{fp1}(\mu)d\mu = t_1 + t_2$. We first calculate t_1 .

$$\begin{aligned} t_1 &= \int_0^1 \frac{d\mu}{\phi_{fp1}(\mu)} = \int_0^1 \frac{d\mu}{\frac{h_1}{p} \mu^{1-p} \exp(\mu^p) + h_2 \mu^q \exp(\mu^{q-1})} \\ &\leq \int_0^1 \frac{p\mu^{p-1} d\mu}{h_1 \exp(\mu^p)} \leq \frac{1}{h_1} \int_0^1 \frac{d\mu^p}{\exp(\mu^p)} \leq \frac{1}{h_1} \left(1 - \frac{1}{\exp(1)}\right). \end{aligned} \tag{14}$$

Next, we calculate t_2 .

$$\begin{aligned} t_2 &= \int_1^{+\infty} \frac{d\mu}{\frac{h_1}{p} \mu^{1-p} \exp(\mu^p) + h_2 \mu^q \exp(\mu^{q-1})} \\ &\leq \int_1^{+\infty} \frac{d\mu}{h_2 \mu^q \exp(\mu^{q-1})} \leq \int_1^{+\infty} \frac{\mu^{2q-2} d\mu}{h_2 \mu^q \exp(\mu^{q-1})} \\ &\leq \frac{1}{h_2(q-1)} \int_1^{+\infty} \frac{d\mu^{q-1}}{\exp(\mu^{q-1})} \leq \frac{1}{h_2(q-1) \exp(1)}. \end{aligned} \tag{15}$$

Substituting t_1 and t_2 into (13), we can know that

$$T_c \leq \frac{1}{\lambda}(t_1 + t_2) \leq \frac{1}{\lambda \exp(1)} \left[\frac{1}{h_1} (\exp(1) - 1) + \frac{1}{h_2(q-1)} \right]. \tag{16}$$

Similarly, for the EPTR-DPZNN model, substituting $\lambda_{adp}(t)$ (8) into Eq. (12) yields

$$\begin{aligned} \frac{d\mu}{dt} &= \frac{d|l(t)|}{dt} = \dot{l}(t) \operatorname{sgn}(l(t)) = -\lambda_{adp}(t) \phi_{fp1}(\mu) \\ &= -\lambda \exp\left(\arctan\left(\frac{\|e(t)\|_F}{\alpha}\right)\right) \phi_{fp1}(\mu) \leq -\lambda \phi_{fp1}(\mu) \\ &\Rightarrow \frac{d\mu}{\phi_{fp1}(\mu)} \leq -\lambda dt \Rightarrow T_c \leq \frac{1}{\lambda} \int_0^{+\infty} \frac{d\mu}{\phi_{fp1}(\mu)}. \end{aligned} \tag{17}$$

The subsequent proofs will follow the progression from Eqs. (14) to (16). Proof is completed.

4.3 Robustness analysis

Theorem 3. *Suppose the EPTR-ZNN and EPTR-DPZNN models are perturbed by an unknown TV-BN or TV-UN matrix $N(t)$, and the elements $n(t)$ of this matrix satisfy the condition $|n(t)| \leq \lambda(h_3|e(t)|^r + h_4)$. Next, the state matrix $X(t)$ of the EPTR-ZNN and EPTR-DPZNN models initiated from an arbitrary initial state $X(0) \in \mathbb{C}^{k \times k}$ can converge to the theoretical solution within the prescribed-time T_c .*

Proof. The evolution formula of the EPTR-ZNN model with noise perturbation are presented below:

$$\dot{E}(t) = -\lambda\left(\Phi_{FP}(E_r(t)) + i\Phi_{FP}(E_m(t))\right) + \Delta N(t). \quad (18)$$

For equation (18), let $\mu = |l(t)|$, where $l(t)$ is defined as in Theorem 1. When $l(t) = 0$, $e_r(t)$ and $e_m(t)$ also converge to zero. According to Theorem 2, we know

$$\begin{aligned} \frac{d\mu}{dt} &= \left[-\lambda\phi_{fp}(l(t)) + \Delta n(t)\right]\text{sgn}(l(t)) \\ &= -\lambda\phi_{fp1}(\mu) - \lambda(h_3\mu^r + h_4) + \Delta n(t)\text{sgn}(l(t)) \\ &\leq -\lambda\phi_{fp1}(\mu) - \left[\lambda(h_3\mu^r + h_4) - |\Delta n(t)|\right]. \end{aligned} \quad (19)$$

Because $|\Delta n(t)| \leq \lambda(h_3|e(t)|^r + h_4)$, the inequality (19) can be depicted as

$$\frac{d\mu}{\phi_{fp1}(\mu)} \leq -\lambda dt \Rightarrow T_c \leq \frac{1}{\lambda} \int_0^{+\infty} \frac{d\mu}{\phi_{fp1}(\mu)}. \quad (20)$$

The subsequent proofs are identical to those from Eqs. (14) to (16).

Similarly, for the EPTR-DPZNN model, substituting $\lambda_{adp}(t)$ (8) into Eq. (19) yields

$$\begin{aligned} \frac{d\mu}{dt} &= -\lambda \exp\left(\arctan\left(\frac{\|e(t)\|_F}{\alpha}\right)\right) \phi_{fp}(\mu) + \Delta n(t)\text{sgn}(l(t)) \\ &\leq -\lambda\phi_{fp1}(\mu) - \left[\lambda(h_3\mu^r + h_4) - |\Delta n(t)|\right]. \\ &\Rightarrow \frac{d\mu}{\phi_{fp1}(\mu)} \leq -\lambda dt \Rightarrow T_c \leq \frac{1}{\lambda} \int_0^{+\infty} \frac{d\mu}{\phi_{fp1}(\mu)}. \end{aligned} \quad (21)$$

The subsequent proofs will follow the progression from Eqs. (14) to (16). Proof is completed.

5 Simulation experiment

To demonstrate the superiority of the EPTR-ZNN and EPTR-DPZNN models in terms of convergence, computational efficiency and robustness. To this end, we compare them with three ZNN models in Table 3 and present the configuration information of the experiments in Table 2.

Table 2. Configuration information of experiments

CPU	GPU	RAM	System	Software	
i7	10870h	RTX 3070	32 GB	Windows 10	Matlab R2021a

Table 3. Comparison of activation functions and convergence parameters of ZNN models.

Model	Activation function	Convergence parameter
PTR-ZNN [9]	$\phi_{ebp}(y) = h_1 \exp(y ^p) y ^{(1-p)} \operatorname{sgn}(y)/p + h_3y + h_4 \operatorname{sgn}(y)$	λ
EPTR-ZNN	$\phi_{fp}(y) = \left(h_1 \exp(y ^p) y ^{(1-p)} + h_2 \exp(y ^{(q-1)}) y ^q \right) \operatorname{sgn}(y) + h_3y^r + h_4 \operatorname{sgn}(y)$	λ
PCCV-ZNN [11]	$\phi_{nsbp}(y) = (h_1 y ^p + h_2 y ^{1/p}) \operatorname{sgn}(y) + h_3y$	$\begin{cases} \lambda \exp(t), 0 < \lambda \leq 1, \\ \lambda^t + 2\lambda t + \lambda, \lambda > 1. \end{cases}$
HTVPR-ZNN [12]	$\phi_{sbp}(y) = (h_1 y ^p + h_2 y ^q) \operatorname{sgn}(y) + h_3y^r + h_4 \operatorname{sgn}(y)$	$\lambda \tanh(e(t) /\alpha)$
EPTR-DPZNN	$\phi_{fp}(y) = \left(h_1 \exp(y ^p) y ^{(1-p)} + h_2 \exp(y ^{(q-1)}) y ^q \right) \operatorname{sgn}(y) + h_3y^r + h_4 \operatorname{sgn}(y)$	$\lambda \exp(\arctan(\ e(t)\ _F/\alpha))$

To verify the validity of the EPTR-ZNN and EPTR-DPZNN models, a solvable TV-PSME was formulated, with the coefficient matrices $O(t)$, $P(t)$, and $Q(t)$ shown as follows:

$$\begin{aligned}
 O(t) &= \begin{bmatrix} 2 + 2iC(2) & C(3) & 4 + iC(1) & 2C(3) - 3iS(2) \\ S(2) + iS(1) & 2S(5) + 4i & 2C(4) - 6i & -6 + 3iS(6) \\ 3C(2) + 2iS(1) & 2S(1) + 2iC(1) & -2 - 3iS(2) & 2C(1) - 2iC(1) \\ -5 - 6iC(5) & -6 - 3S(3) & 4 + 7iS(2) & 3C(3) + 3iS(3) \end{bmatrix}, \\
 P(t) &= \begin{bmatrix} 4 + iC(2) & 2S(3) & 2 - 2iS(2) & -6C(3) + 3i \\ 2S(3) + iC(2) & 4C(3) - 2iC(4) & -4S(4) + 4i & -4 - 2iS(4) \\ 4S(3) - 4i & -5 - 2iC(2) & 3 + 4iC(3) & 4C(4) + 4iS(4) \\ -6C(4) + 3i & 3 + 2iS(2) & 2C(8) - 4i & 3S(2) - 4iC(3) \end{bmatrix}, \\
 Q(t) &= \begin{bmatrix} -3 + iC(1) & 3S(1) - 5iC(2) & -5S(4) + 6iC(3) & 8 - 4iS(4) \\ C(4) - 4i & -4S(1) + 2iC(1) & 7 + 2iS(2) & -5C(2) + 5S(2) \\ -3 + 3S(1) & 5 + 2iS(7) & 3C(6) - 5i & 3S(3) - 3i \\ 3S(3) - 3iC(2) & -2 + 3iS(2) & 4 - 4iC(4) & -4S(1) + 2i \end{bmatrix},
 \end{aligned}$$

where $S(n) = \sin(nt)$, $C(n) = \cos(nt)$ and n is an integer.

Firstly, in a noiseless environment, we assess the convergence performance of ZNN models under consistent parameter settings for CPs and AFs. When $\lambda = r = 1$, $\alpha = 0.01$, $p = 0.6, q = 1.4, h_1 = h_2 = 2$ and $h_3 = h_4 = 0$, the results are presented in Figs. 1 and 2. Fig. 1 indicates that while all models reach the theoretical solution $X^*(t)$ within 2 seconds, their convergence rates vary significantly. Fig. 2(a) shows more detailed information. By Theorem 2, the EPTR-ZNN and EPTR-DPZNN models' theoretical prescribed convergence time T_c is 0.78 seconds. However, they converge in 0.36 and 0.08 seconds respectively, exceeding this expectation and significantly outperforming other models. To validate the generality of the conclusions, we vary the parameters p and α in the same initial state $X(0)$ and let $p = 0.4$ and $\alpha = 0.001$. The results are shown in Fig. 2(c). The same conclusion can be obtained in Fig. 2(c), i.e., the actual convergence time of the EPTR-ZNN and EPTR-DPZNN models is much lower than their theoretical prescribed convergence time T_c , and their convergence rate is better than the other models.

To compare the computational efficiency of ZNN models, we obtained Table 4. The T_{CP} of ZNN models is proportional to CC. Table 4 reveals that the EPTR-ZNN and EPTR-DPZNN models exhibit faster T_{CV} , lower T_{CP} and CC compared to other ZNN models. The EPTR-DPZNN model, incorporating the ADP $\lambda_{adp}(t)$, further enhances the convergence performance and computational efficiency compared to EPTR-ZNN model. In addition, by comparing the two

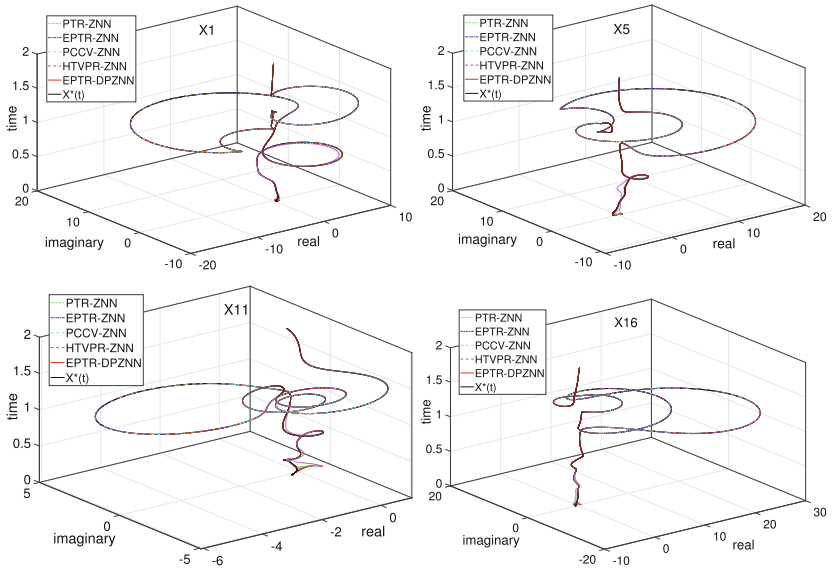


Fig. 1. Part of state solution $X(t)$ and part of theoretical solution $X^*(t)$ of ZNN models

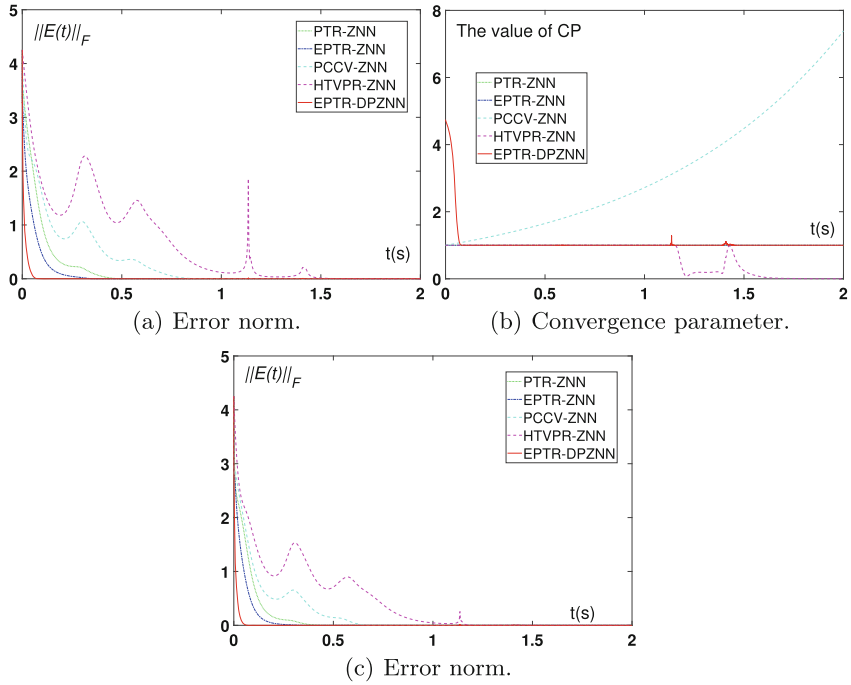


Fig. 2. The error norm and convergence parameter obtained by ZNN models.

sets of data in Table 4, it can be seen that the T_{CP} and CC for the PCCV-ZNN and HTVPR-ZNN models increase as the T_{CV} decreases. In contrast, the T_{CP} and CC of the EPTR-ZNN and EPTR-DPZNN models decrease simultaneously with the decrease of T_{CV} , which means that their convergence performance and computational efficiency are both improved.

Table 4. Comparison of computational efficiency of ZNN models. The values of T_{CP} and CC are calculated in the convergence phase of the model.

Parameter settings	Model	Convergence time (T_{CV})	Computational time (T_{CP})	Computation counts (CC)
Fig. 2(a):	PTR-ZNN [9]	0.50 s	58.19 s	656
$\lambda = r = 1, \alpha = 0.01,$	EPTR-ZNN	0.36 s	38.80 s	488
	PCCV-ZNN [11]	0.93 s	39.67 s	492
$p = 0.6, q = 1.4,$	HTVPR-ZNN [12]	1.74 s	85.61 s	1076
$h_1 = h_2 = 2,$	EPTR-DPZNN	0.08 s	29.14 s	362
$h_3 = h_4 = 0.$				
Fig. 2(c):	PTR-ZNN [9]	0.48 s	34.79 s	386
$\lambda = r = 1, \alpha = 0.001,$	EPTR-ZNN	0.35 s	24.77 s	314
	PCCV-ZNN [11]	0.67 s	64.46 s	740
$p = 0.4, q = 1.4,$	HTVPR-ZNN [12]	1.22 s	106.17 s	1244
$h_1 = h_2 = 2,$	EPTR-DPZNN	0.074 s	21.21 s	248
$h_3 = h_4 = 0.$				

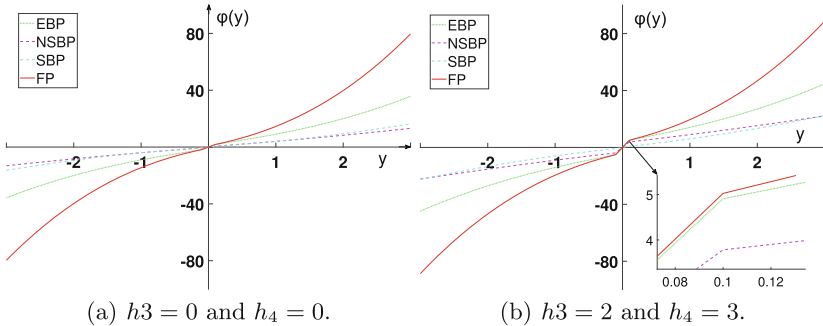


Fig. 3. The activation functions of ZNN models.

Secondly, to highlight the advantages of ADP $\lambda_{adp}(t)$ and AF $\phi_{fp}(\cdot)$, we compared the variation trends of CPs and AFs from different ZNN models in Figs. 2(b) and 3. From Fig. 2(b), it is apparent that $\lambda_{adp}(t)$ of the EPTR-ZNN model diminishes as the model converges, eventually stabilizing at 0.08 seconds. This is consistent with the convergence time of EPTR-DPZNN model as depicted in Fig. 2(a). However, The CPs of non-HTVPR-ZNN models remain constant

or increase over time, with PCCV-ZNN showing an increase. Despite HTVPR-ZNN's CP decreasing, it eventually reaches to 0, which is not conducive to the stability of model after convergence. Moreover, Figs. 3 and 2(a) show a positive correlation between model's convergence rate and the slope of $\phi(\cdot)$, indicating the superior convergence performance of the $\phi_{fp}(\cdot)$.

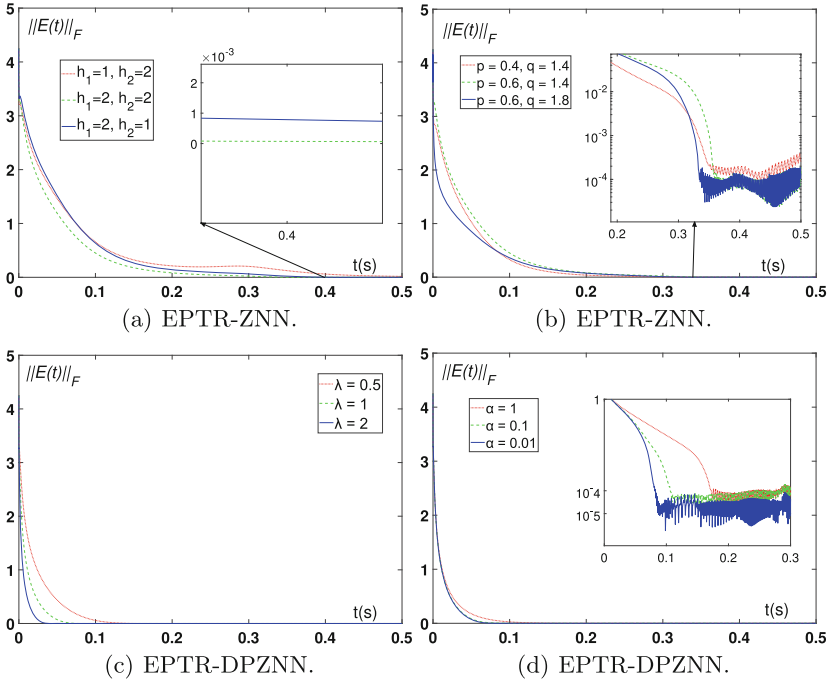


Fig. 4. Impact of parameters on the convergence rate of ZNN models.

Thirdly, using the control variables method, we investigate the effect of CPs and AFs on the convergence rate of EPTR-ZNN and EPTR-DPZNN models. The results are shown in Fig. 4. As λ , q increase and p , β decreases, the convergence rate of the EPTR-ZNN and EPTR-DPZNN models improves. That is, as the values of CP and AF increase, the ZNN model converges faster.

Finally, to explore the robustness of ZNN models, we let $\Delta N(t)$ be the noise matrix representing the external noise with its elements as $\Delta n(t)$ and let $h_3 = 0.2$, $h_4 = 0.3$, and the other parameters are the same as in Fig. 2(a). We introduce two different types of noise, time-variant bounded noise (TV-BN) and time-variant unbounded noise (TV-UN), exceeding the limit, i.e., $|\Delta n(t)| > \lambda(h_3|e_{ij}(t)|^r + h_4)$. Fig. 5(a)-(b) shows the experimental results. Except for the EPTR-DPZNN model, the errors of the other ZNN models do not converge stably to 0. Next, to ensure the generality of conclusions, we let $h_3 = 2$, $h_4 = 3$, and increase the amplitude of noise, where $|\Delta n(t)| \leq \lambda(h_3|e_{ij}(t)|^r + h_4)$.

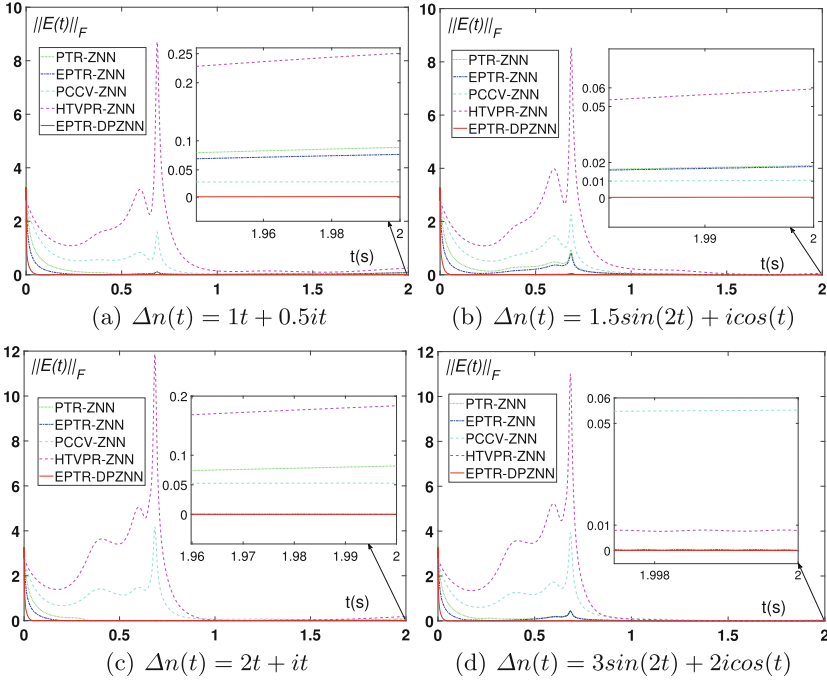


Fig. 5. Error norm of ZNN models in noisy environment.

The experimental results are shown in Fig. 5(c)-(d). After increasing the noise amplitude, the errors of PCCV-ZNN and HTVPR-ZNN models still fail to converge, while the PTR-ZNN model converges only in the case of TV-BN. The errors of EPTR-ZNN and EPTR-DPZNN models could converge to 0, but the EPTR-DPZNN model had smaller error fluctuations. The above analysis of convergence, computational efficiency and robustness can prove that our design philosophy of the ADP $\lambda_{adp}(t)$ and AF $\phi_{fp}(\cdot)$ is correct.

6 Conclusions

In this paper, the EPTR-ZNN and EPTR-DPZNN models are proposed to compute the TV-PSME. The EPTR-ZNN model has a novel AF. Unlike the FP used in the EPTR-ZNN model, the EPTR-DPZNN model designed on its basis adopts a novel ADP. The novel AF improves the convergence rate and robustness of the model. The ADP not only improves the convergence performance and robustness of the model but also enhances its computational efficiency. Theoretical analysis proves the prescribed time convergence and robustness of all models. Simulation experiments demonstrate the superior convergence performance, efficiency, and robustness of EPTR-ZNN and EPTR-DPZNN models. Future research will be directed towards the further enhancement of the computational efficiency of these two ZNN models and their subsequent application to practical areas.

References

1. Chong, E.K., Hui, S., Zak, S.H.: An analysis of a class of neural networks for solving linear programming problems. *IEEE Trans. Autom. Control* **44**(11), 1995–2006 (1999)
2. Huang, Y., Chen, J., Huang, L., Zhu, Q.: Dynamic games for secure and resilient control system design. *Natl. Sci. Rev.* **7**(7), 1125–1141 (2020)
3. Jin, L., Yan, J., Du, X., Xiao, X., Fu, D.: Rnn for solving time-variant generalized sylvester equation with applications to robots and acoustic source localization. *IEEE Trans. Industr. Inf.* **16**(10), 6359–6369 (2020)
4. J. Jin, J. Zhu, J. Gong, and W. Chen, “Novel activation functions-based znn models for fixed-time solving dynamirc sylvester equation,” *Neural Computing and Applications*, vol. 34, no. 17, pp. 14 297–14 315, 2022
5. Li, W.: A recurrent neural network with explicitly definable convergence time for solving time-variant linear matrix equations. *IEEE Trans. Industr. Inf.* **14**(12), 5289–5298 (2018)
6. Zhang, Y., Ge, S.S.: Design and analysis of a general recurrent neural network model for time-varying matrix inversion. *IEEE Trans. Neural Networks* **16**(6), 1477–1490 (2005)
7. Li, S., Chen, S., Liu, B.: Accelerating a recurrent neural network to finite-time convergence for solving time-varying sylvester equation by using a sign-bi-power activation function. *Neural Process. Lett.* **37**, 189–205 (2013)
8. Xiao, L., Li, L., Tao, J., Li, W.: A predefined-time and anti-noise varying-parameter znn model for solving time-varying complex stein equations. *Neurocomputing* **526**, 158–168 (2023)
9. Dai, J., Jia, L., Xiao, L.: Design and analysis of two prescribed-time and robust znn models with application to time-variant stein matrix equation. *IEEE transactions on neural networks and learning systems* **32**(4), 1668–1677 (2020)
10. Xiao, L., He, Y., Dai, J., Liu, X., Liao, B., Tan, H.: A variable-parameter noise-tolerant zeroing neural network for time-variant matrix inversion with guaranteed robustness. *IEEE Transactions on Neural Networks and Learning Systems* **33**(4), 1535–1545 (2020)
11. Xiao, L., Tao, J., Dai, J., Wang, Y., Jia, L., He, Y.: A parameter-changing and complex-valued zeroing neural-network for finding solution of time-varying complex linear matrix equations in finite time. *IEEE Trans. Industr. Inf.* **17**(10), 6634–6643 (2021)
12. Luo, J., Yang, H., Yuan, L., Chen, H., Wang, X.: Hyperbolic tangent variant-parameter robust znn schemes for solving time-varying control equations and tracking of mobile robot. *Neurocomputing* **510**, 218–232 (2022)
13. Z. Qi, Y. Ning, L. Xiao, Z. Wang, and Y. He, “Efficient predefined-time adaptive neural networks for computing time-varying tensor moore–penrose inverse,” *IEEE Transactions on Neural Networks and Learning Systems*, 2024



HRA: Heuristic Reordering Approach for Preserving Dependency in Hierarchical Time Series Forecasting

Santosh Palaskar¹(✉), Surya Shravan Kumar Sajja², Nandyala Hemachandra¹,
and Narayan Rangaraj¹

¹ IIT, Bombay, India

{santoshpalaskar77,nh,narayan.rangaraj}@iitb.ac.in

² IBM Research, Bengaluru, India

suryasku@in.ibm.com

Abstract. Hierarchical time series analysis requires probabilistic forecasting techniques to account for inherent uncertainties. A probabilistic forecast proposes a range of potential outcomes. In domains like retail and electricity, where time series data exhibit significant cross-correlations and multiple hierarchical levels, existing research has not emphasized the development of models that consider these dependencies. This lack of attention is mainly due to the recently reported good performance of the simpler independent models. In response to this challenge, we introduce HRA (Heuristic Reordering Approach), a novel approach designed to enhance predictive accuracy and preserve the dependencies. Notably, HRA does post-processing using a heuristic recording technique on forecasted values and is adaptable to samples of any size. Our detailed experiments demonstrate the effectiveness of HRA by improving accuracy by up to 7% compared to existing state-of-the-art (SoTA) methods on simulated and well-established benchmark datasets. These results underscore HRA's ability to significantly improve forecasting accuracy, preserve the correlation and address the unique complexities associated with hierarchical time series data.

Keywords: Time Series · Probabilistic Hierarchical Forecast · Sample Reordering · Coherence · Rank Correlation

1 Introduction

Previous research has predominantly focused on point forecasts, neglecting the importance of probabilistic forecasting. Distinguishing between risks and opportunities is often hindered by the inherent uncertainty in various factors such as

Part of this work was done during an internship in IBM Research, India.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_25.

production rates, inventory levels, lead times, safety stocks, etc. Relying solely on deterministic forecasts makes it challenging to formulate effective improvement plans. Probabilistic forecasts allow us to understand the uncertainty associated with forecasts, preventing us from making decisions based on false beliefs and expectations. Time series data often follows a hierarchical aggregation structure. In hierarchical time series, we have a hierarchy of time series, i.e., time series at different levels that depend on each other. While forecasting, we can aggregate (based on the parent-child relationship) lower-level time series to get a forecast for higher levels, and in aggregation, coherence should be maintained. For example, in retail supply chains, periodic demand for various levels, like retail stores, can roll up to different geographical hierarchies in a city, state and country. These aggregated demand streams are examples of hierarchical time series. Hierarchical time series is difficult to forecast because we need to model multiple loosely correlated time series while maintaining coherence across different levels of hierarchy. Coherence implies that lower-level forecasts add up to the aggregated forecast.

For hierarchical time series forecasting, we encounter uncertainties related to data, model specifications and parameter estimates. Therefore, a shift towards probabilistic forecasting becomes imperative when dealing with hierarchical structures. Authors in [10] highlighted two prominent domains that require probabilistic and hierarchical forecasting: 1) Retail demand forecasting and 2) Electricity demand forecasting. The retail industry relies heavily on demand forecasting to optimize their supply chains. Probabilistic forecasts play a critical role in enhancing decision-making about inventory, safety stock, lead times and supply chain operations. These decisions span various supply chain levels, from individual stores to regional and national scales. Probabilistic forecasts enable retailers to estimate the likelihood of stockouts and take proactive measures to improve customer satisfaction while reducing costs. Effective management of energy resources hinges on accurate electricity demand forecasting. Hierarchical forecasting involves decision-making at multiple levels of an electrical grid. Probabilistic forecasts are indispensable because many factors, such as weather conditions, economic variables and unforeseen events, influence electricity demand. Stakeholders, including utility companies, can optimize power generation, distribution, and grid management by leveraging probabilistic forecasts.

1.1 Related Work

Forecasting within a hierarchical structure poses significant challenges due to the diverse interactions and varying levels of data aggregation across the hierarchy. Lower-level time series data often exhibit noise, making autoregressive models less effective. Aggregation results in smoother time series are suitable for autoregressive models as we move up the hierarchy. However, generating independent forecasts for each series, called the base forecast, may not yield a coherent forecast. This issue becomes particularly complex when dealing with data consisting of many time series. Probabilistic forecasting plays a crucial role in addressing uncertainty, particularly for unpredictable, slow-moving, long-tail Stock Keeping Units (SKU) or those with limited order history. A probabilistic forecast quantifies uncertainty and facilitates improved decision-making and risk management.

Authors in [10] have proposed an innovative algorithm for generating predictive distributions with correlated samples within the hierarchy. This algorithm reorders samples based on ranks of past data residuals, considering varying mean and standard deviation for each series while maintaining a constant conditional distribution over standard residuals. This approach ensures the coherence and reliability of probabilistic forecasts.

In the domain of probabilistic forecasting, various methodologies have been proposed to address different challenges. While some models focus on flexibility and coherence through distributional coherency regularization, others aim for coherent hierarchical forecasts without additional post-processing. However, these approaches do not prioritize preserving cross-correlations within and across hierarchical levels [6]. Several studies from [7, 11] have demonstrated that data with high correlation, when subjected to models that explicitly account for the temporal correlation, tend to yield sub-optimal performance compared to models that perform independent forecast. Some models have attempted to capture this correlation explicitly, but their performance falls short compared to models assuming independence.

Given these observations, we propose a novel approach called Heuristic Reordering Approach (HRA) for preserving correlation in hierarchical time series forecasting. This model combines two key stages: i) Independent base forecast: generates forecasts independently for each time series. ii) Reordering with HRA: following the initial independent forecasts, the HRA model is applied to reorganize the forecasts while preserving the inherent correlation within the data. It minimizes the objective of correlation difference by reordering the p fraction of Extreme Values.

1.2 Summary of Contributions

- We introduced a novel Heuristic Reordering Approach (HRA) to preserve dependency among multiple time series while maintaining coherence across the hierarchical levels.
- HRA demonstrates superior performance to state-of-the-art (SoTA) methods. This is validated through extensive evaluations on simulated data as well as established benchmark datasets, including *Tourism* [2], *Wiki* [6], and *Labour* [1] (Section 4).
- For forecast horizons H from existing studies, we evaluated performance at $2H$ and $3H$ to assess forecasting accuracy over longer timeframes.
- We comprehensively analyse how HRA outperforms existing SoTA methods in preserving correlation; detailed analysis is presented in Section 4.2.
- We also conducted an ablation study to show that a small fraction p of Extreme Values is sufficient to preserve the correlation (Section 4.3).

2 Hierarchical Probabilistic Forecasting

Table 1. Useful Notation

n	total number of time series in the hierarchy
r	total number of aggregated series
m	total number of bottom level series
K	number of samples extracted from the distribution
T	total time periods
$\hat{Y}_{i,T+h}$	predicted values for time series i for horizon h , $i = 1, \dots, n, h = 1, \dots, H$
$\rho(Y_i, Y_j)$	Spearman's rank correlation coefficient between series Y_i and Y_j
$\hat{F}_{i,t}$	condition predictive distribution at time t for time series i , given by forecasting model
$i(n_w)$	w^{th} child node of aggregated series i ; $w \in 1, \dots, n_c$
p	fraction of Extreme Values to be selected from a sample set.

A hierarchical time series is a collection of time series that follows a hierarchical aggregation structure. Let $\mathbf{a}_t \in \mathbb{R}^r$ be a vector containing observations at different levels of aggregation at time t , $\mathbf{b}_t \in \mathbb{R}^m$ be a vector containing observations at the bottom level of hierarchy. Now let $\mathbf{y}_t = (\mathbf{a}_t, \mathbf{b}_t)'$ be a vector of size $r + m = n$, that contains all the observations in the hierarchy. We can write $\mathbf{y}_t = \mathbf{S}\mathbf{b}_t$, where $\mathbf{S} = [\mathbf{S}'_a \ \mathbf{I}_m]' \in \{0, 1\}^{n \times m}$ is the summing matrix, $\mathbf{S}_a \in \{0, 1\}^{n \times m}$ and \mathbf{I}_m is an identity matrix of order m . We assume that we have access to T historical observations of \mathbf{y}_t , and it is possible to forecast time series at all levels independently. We can estimate $\mathbb{E}[y_{i,T+h}|y_1, y_2, \dots, y_{T-1}, y_T]$ for $i = 1, \dots, n$ and $h = 1, \dots, H$. These are called *base* forecasts, and they do not always satisfy *coherence* constraints, i.e., forecasts do not add up in the same way as the data. Over the past few decades, the forecasting literature has experienced a notable shift towards hierarchical forecasting [5]. This shift reflects a continuous effort to enhance classical hierarchical forecasting methods, such as bottom-up and top-down approaches [5]. These improvements have been mainly focused on extending these methods to accommodate probabilistic forecasting, enabling uncertainty quantification, more informed decision-making and effective risk management. Quantifying uncertainty should not only rely on the conditional mean and variance. We can introduce the concept of a conditional predictive cumulative distribution function (CPCD) [10]. This distribution function can be represented as follows:

$$F_{i,T+h}(y|\mathbf{y}_1, \dots, \mathbf{y}_T) = P(y_{i,T+h} \leq y | \mathbf{y}_1, \dots, \mathbf{y}_T) \quad (1)$$

with $i = 1, \dots, n$. CPCD provides a more comprehensive understanding of uncertainty. It captures the entire conditional distribution, allowing us to assess the likelihood of various outcomes and make more informed decisions. Probabilistic forecasts for individual series can be computed independently. However, these forecasts may not necessarily be coherent. Hierarchical probabilistic forecasts are considered coherent when the predictive distribution of each aggregate series matches the distribution of the sum of the children series. In our setup, we used the bottom-up method. Bottom-up produces the samples of each aggregated

series using only the predictive distribution of the bottom series; hence, it is coherent by construction [10].

Algorithm 1 HRA: Heuristic Reordering Approach for Bottom-Up Probabilistic Forecasting

1. Input $Y_1, Y_2 \dots, Y_n, p \in (0, 1)$
2. For all bottom level series $i = r + 1, \dots, n$, fit the individual probabilistic models to calculate:
 - (a) Conditional distribution $\hat{F}_{i,T+h}$ for each future period $h = 1, \dots, H$ as shown in Equations (1) and (3).
 - (b) Extract a sample of size K , say x_1^i, \dots, x_K^i from $\hat{F}_{i,T+h}$, where K samples are the Monte Carlo samples describing the joint distribution over time and components.
3. For all aggregated series $i = 1, \dots, r$:
 - (a.) Let $C_i = \{i(1), \dots, i(n_c)\}$ be the set of n_c child nodes of aggregate node i .
 - (b.) For all child node tuples $(\ell, x), \ell \in C, x \in \ell^c, \ell^c = C_i \setminus \{\ell\}$:
 - i. Let $\rho(\ell, x)$ represent sample Spearman’s rank correlation between time series ℓ and x (using past data values).
 - ii. Let $\hat{\rho}_{t+h}(\ell, x)$ represent predicted sample Spearman’s rank correlation between time series ℓ, x (using forecasts at time $t + h, h = 1, \dots, H$).
 - iii. Let $D_{t+h}(\ell, \ell^c) = \sum_{x \in \ell^c} |\rho(\ell, x) - \hat{\rho}_{t+h}(\ell, x)|$ represent the correlation difference between the child node ℓ and remaining child nodes $C_i \setminus \{\ell\}, h = 1, \dots, H$.
 - iv. For all $t + h, h = 1, \dots, H$:
 - Take $\frac{p}{2}$ fraction of max and $\frac{p}{2}$ fraction min values (Extreme Values) of series ℓ and let it be represented as $\mathcal{L}_{p,T+h}$.
 - For all values in $\mathcal{L}_{p,T+h}$, follow the steps given in the Algorithm 2.
 - (c.) Recursively compute:

$$x_k^i = x_k^{i(1)} + \dots + x_k^{i(n_c)}$$

Where x_k^i denote the k^{th} sample of aggregated series i at time step $T + h$.

4. Output: $\hat{Y}_{1,T+h}, \hat{Y}_{2,T+h} \dots, \hat{Y}_{n,T+h}$.
-

2.1 Bottom-up Probabilistic Forecasting with Heuristic Reordering

Algorithm 2 Heuristic Reordering

1. Let $\mathcal{L}_{p,T+h}$ be the set of p fraction of total forecasted sample values as Extreme Values of ℓ with $p \in (0, 1)$.
2. For $e = 1, \dots, N_{epoch}$:
 - For all $x_{j,T+h} \in \mathcal{L}_{p,T+h}$:
 - (a) Swap position of $x_{j,T+h}$ with remaining samples $\{x_{1,T+h}, \dots, x_{K,T+h}\} \setminus \{x_{j,T+h}\}$.
 - (b) With every swap track the objective $D_{t+h}(\ell, \ell^c)$:
 - If, objective improves (minimizes), save it as the best correlation difference and store the position of $x_{j,T+h}$.
 - Else, no change in objective
 - (c) After recursive swapping over all samples, fix the position of $x_{j,T+h}$ at the location where we get the least correlation difference.
 - (d) Follow the same steps for all the $x_{j,T+h} \in \mathcal{L}_{p,T+h}$ except in step (c) we cannot swap with the position fixed for earlier $x_{j,T+h}$.

Select the epoch with the lowest correlation score, assign the best epoch positions to the Extreme Values and save it.

In hierarchical time series forecasting, it is crucial to maintain coherence across the levels of the hierarchy. Thus, we employ the classical bottom-up method [5], which is coherent by construction. Algorithm 1 uses a bottom-up probabilistic approach that reorders the predicted samples at each period to preserve the correlation across the time series under the same parent node. Firstly, we generate independent base forecasts for each of the m bottom series. Then, we apply the HRA as an alternative to the residual rank reordering method proposed for probabilistic forecasting by [10]. In residual rank reordering, we find the standard residual (\hat{e}_{it}) and permutations ($P_i(t)$) on the predicted values for the past data, defined as $\hat{e}_{it} = (y_{i,t} - \hat{\mu}_{i,t})/\hat{\sigma}_{i,t}$ and $P_i(t) = rk(\hat{e}_{it})$. $\mu_{i,t}$ and $\sigma_{i,t}$ are the predicted mean and standard deviation for the time series i at time t . $rk(\hat{e}_{it})$ is the rank of the standard residual for time series i at time t . Predicted samples for each time step equal in size to past data are reordered according to the permutation defined on the past data. This reordering preserves the rank correlation across the time series under the same aggregated node.

HRA does not require a sample size equal to the past forecast length because data points are often not enough to define the distribution. Instead, we can generate sufficient samples for future periods and employ HRA to reorder them. We define the Extreme Values of each time period in the forecast horizon; **Extreme Values** are defined as the p fraction of the total forecasted sample set. They consist of $\frac{p}{2}$ minimum and $\frac{p}{2}$ maximum values from the total sample set. Experiments showed that Extreme Values greatly impact the aggregated forecast, so an

appropriate reordering of Extreme Values should result in better predictive distribution for the aggregated series. In reordering, we swap each Extreme Value with the remaining samples for each time step by continuously monitoring the objective. We identify the optimal Extreme Value position by keeping it fixed at the point where the minimum objective value is achieved. The objective function, as defined in Algorithm 1, 3(b)iii, is the sum of the difference between the correlation between the selected child node and other remaining child node past and predicted values. Each aggregated level time series forecast is computed using the bottom-up method on reordered samples. HRA outperforms the SoTA methods by a significant margin and effectively preserves the dependence between the past and forecasted values of the time series; detailed results are given in the experiments and results (Section 4).

2.2 Base Probabilistic Forecasting Methods

In this paper, we employ:

- DeepAR [9] is a method based on autoregressive recurrent networks. It uses LSTM-based recurrent neural networks to generate probabilistic forecasts.
- DLinear [11] is a set of simple linear models called LTSF-Linear. These linear models outperform complex Transformer-based models on real-world datasets, highlighting the potential limitations of Transformers in capturing temporal information in time series.
- NLinear [11] is a basic one-layer linear model used to assess Transformer-based approaches for long-term time series forecasting (LTSF). Unlike Transformers, which employ self-attention for semantic correlation extraction, NLinear relies on simple linear transformations to capture temporal relations in time series data.
- N-HiTS [3] is a neural network based model that applies hierarchical interpolation and multi-rate data sampling to take the volatility of the predictions and their computational complexity.
- Lag-Llama [8] is a transformer-based model for univariate probabilistic time series forecasting, pretrained on diverse datasets for strong zero-shot generalization. It achieves state-of-the-art performance when fine-tuned on small fractions of previously unseen data, emerging as the best general-purpose model on average.

The output of these models will be probabilistic; In local models like AutoRegressive Integrated Moving Average (ARIMA) and Exponential Smoothing, Monte Carlo sampling generates distributions of future values, capturing forecast uncertainty. Similarly, neural networks use a likelihood function during training to model the underlying data distribution. This allows the network to produce probabilistic forecasts by sampling from the learned distribution.

2.3 Probabilistic Forecast Measure

Continuous Rank Probability Score (CRPS) [4] is a scoring rule to evaluate probabilistic predictions or forecasts by comparing them with ground truth values

and generating a single comparable value. CRPS compares a single ground truth value to a Cumulative Distribution Function F :

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - \mathbb{1}\{y \geq x\})dy \tag{2}$$

In this paper, we use the empirical CDF to calculate the CRPS given as

$$\hat{F}(x) = \frac{1}{K} \sum_{k=1}^K \mathbb{1}\{X_k \leq x\} \tag{3}$$

where X_1, \dots, X_n are a sample of size n from a population with CDF $F(x)$.

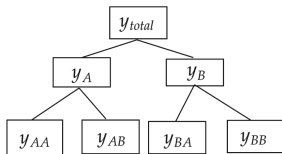
We can also use the weighted form of CRPS, which puts more emphasis on probability levels of greater interest. Given h period-ahead predictive distribution \hat{F}_{t+h} and an observation y_{t+h} , the quantile-weighted version of the CRPS is

$$\text{Weighted CRPS} \left(\hat{F}_{t+h}, y_{t+h} \right) = \int_0^1 v(\tau) \text{QS}_{\tau} \left(\hat{F}_{t+h}^{-1}(\tau), y_{t+h} \right) d\tau$$

where $v(\tau)$ is a non-negative weight function on the unit interval, and QS_{τ} is the quantile score at probability level τ , defined as

$$\begin{aligned} \text{QS}_{\tau} \left(\hat{F}_{t+h}^{-1}(\tau), y_{t+h} \right) = \\ 2 \left(\mathbf{1} \left\{ y_{t+h} \leq \hat{F}_{t+h}^{-1}(\tau) \right\} - \tau \right) \left(\hat{F}_{t+h}^{-1}(\tau) - y_{t+h} \right) \end{aligned}$$

When closed-form expressions for evaluating an expression are not available, a discretized approximate version can be computed to any degree of accuracy. Skill-CRPS is defined as $1 - \frac{CRPS_{model}}{CRPS_{base}}$. Here, $CRPS_{base}$ represents the CRPS calculated for the base forecast, while $CRPS_{model}$ represents CRPS for alternative or improved forecasting models (e.g., RBU, HRA, etc.). These models are compared to the baseline (base forecast). It is important to note that a lower CRPS value signifies a more accurate forecast. Therefore, in accordance with the Skill-CRPS definition, a higher Skill-CRPS is preferred. Both $CRPS_{model}$ and $CRPS_{base}$ are computed as the average CRPS values across all observations within the test dataset.



(a) Simulated time series hierarchy

$$\begin{aligned} \mu &= [5, 10, 7, 8] \\ \Sigma &= \begin{bmatrix} 2 & 0.8 & 0.5 & 0.2 \\ 0.8 & 3 & 0.22 & 0.23 \\ 0.5 & 0.22 & 3 & 0.7 \\ 0.2 & 0.23 & 0.7 & 3 \end{bmatrix} \end{aligned}$$

(b) Mean and Correlation matrix for errors

Fig. 1. Three-level time series hierarchy with total number of series $n = 7$ and with mean μ and correlation matrix Σ .

3 Datasets

We compare the performance of probabilistic forecasting models on both simulated and benchmark datasets: *Tourism*, *Wiki*, and *Labour*. A summary of the datasets is given in Table 2. The simulated data is generated using the ARIMA process. These datasets form a hierarchy of seven series ($n = 7$) with three aggregated series ($r = 3$) and four at the bottom level ($m = 4$). At the lowest level, each series is generated using an ARIMA process with a linear trend, with parameters p and q selected randomly from 0, 1, 2. The errors for the bottom-level series $y_{AA}, y_{AB}, y_{BA}, y_{BB}$ are drawn from a multivariate Gaussian distribution, considering strong correlations between series with the same parent and moderate to small correlations between those with different parents (see Fig. 1b). Each time series is simulated for $T = 300$, with 10, 20, and 30 data points used as test points.

Table 2. Summary of the hierarchy for the datasets; H: Horizon, Freq: Frequency

Datasets	Total	Bottom	Freq	H	Levels
<i>Simulated</i>	7	4	Daily	10, 20, 30	3
<i>Tourism</i>	89	56	Quarterly	2, 4, 6	4
<i>Wiki</i>	199	150	Daily	7, 14, 21	5
<i>Labour</i>	57	32	Month	8, 16, 21	4

Tourism, The Australian domestic tourism dataset [2] pertains to the individuals who travelled to Australia for diverse reasons from January 1998 until October 2006, focusing on quarterly data for visitors with purposes like Holiday, Visiting Friends and Relatives, Business, and Other. The *Wiki* dataset [6] compiles daily views of 145,000 Wikipedia articles grouped into 150 categories. The *Labour* dataset [1] provides monthly employment statistics in Australia from February 1978 to December 2020, categorized by employment type, gender, and region (across 8 distinct regions).

4 Experiments and Results

We assess the enhancement of our HRA over SoTA techniques, such as DLinear, NLinear, DeepAR, Lag-Llama and NHiTS. Initially, we generate forecasts using these models (base forecast). Subsequently, we refine the forecasts using reordering techniques outlined below

1. Base Forecast (BASE): this is the initial probabilistic forecast provided by the model without any reordering.
2. Bottom-Up Forecast (BU): forecast is computed by aggregating the predictions from the lower-level child nodes by considering the aggregation structure (hierarchy) [5].

3. Revised Bottom-Up Forecast (RBU): forecast from the lower-level child nodes is summed after being reordered based on the ranks of the residuals [10].
4. Heuristic Reordering Approach (HRA): the aggregated forecasts are calculated using a bottom-up method to re-ordered samples from the lower-level child nodes, detailed in Algorithm 1.

The goal is to improve the accuracy and preserve the correlation among the models (DLinear, NLinear, and Lag-Llama) that perform multivariate forecasting independently. Additionally, we also verify whether HRA can maintain the correlation for models (DeepAR, NHiTS) that already consider dependence. The code for the detailed experiments for all the datasets and models can be accessed at: <https://github.com/santoshpalaskar77/HRA>

4.1 Accuracy Improvements

Simulated data is generated using the ARIMA process, with the parameters p and q chosen randomly, introducing variability into the data generation process. We conducted 5 independent runs to mitigate this randomness and averaged the results for simulated and benchmark datasets. For each dataset, we selected a forecast horizon H from existing studies and additionally evaluated the performance at $2H$ and $3H$ to assess forecasting accuracy over longer timeframes. Due to page size constraints, we present the results in Table 3 up to a test length of $2H$. A detailed comparison is available in Table 1 and Table 2 of the supplementary material. We forecasted 50 samples for each time point within the forecast horizon. Therefore, the past correlation is defined based on the most recent 50 past values. For the tourism dataset, we use a batch size of 16 due to its smaller size, while for the other three datasets, we use a batch size of 32. We set $p = 0.3$ across all datasets and models. All the models were experimented on a GeForce RTX-4090 GPU.

We evaluate the advantages of HRA over state-of-the-art (SoTA) methods using CRPS and Weighted CRPS. Table 3 presents a detailed comparison. The base forecast employs SoTA forecasting models, including DLinear, NLinear, DeepAR, Lag-Llama and NHiTS. Subsequently, bottom-up and reordering models are applied to the base forecast to enhance the probabilistic forecast further. In Table 3, both RBU and HRA consistently outperform the baseline (base forecast without reordering) across almost all scenarios. We also compared the improvements of HRA over the second best performing model from Base, BU, and HRA for each dataset. We observed up to 7% improvement in CRPS and weighted CRPS. HRA performs superior to RBU across all base forecasting methods. However, in the case of NHiTS and Lag-Llama for the *Wiki* dataset, HRA exhibits a slight underperformance compared to the Base forecast yet still outperforms BU and RBU forecasts. This underperformance can be attributed to the characteristics of the *Wiki* dataset. For *Wiki* data, most of the aggregated series has only one child node, limiting the improved performance of reordering-based algorithms. As HRA relies on child node reordering, it exhibited a marginal

Table 3. Comparing HRA improvements over SoTA methods. Comparison is made across base forecasts generated by the SoTA model, with bottom-up and reordering techniques applied over the base forecasts. The best and second-best results are highlighted in **bold** and underlined, respectively. Performance is evaluated using Weighted CRPS, where lower loss values indicate better probabilistic forecasts. The interval for % improvement of HRA over best of Base, BU, RBU is given for all datasets. HRA gives improvement up to 7%.

Data		Simulated		Labour		Wiki		Tourism	
Model	Test len	10	20	8	16	7	14	2	4
DLinear	Base	0.0365	0.0526	0.0462	0.0232	0.4878	0.6724	0.0810	0.0983
	BU	0.0367	0.0531	<u>0.0417</u>	<u>0.0225</u>	0.4452	<u>0.6484</u>	0.0772	0.0954
	RBU	<u>0.0361</u>	<u>0.0522</u>	0.0418	0.0229	<u>0.4443</u>	0.6505	<u>0.0763</u>	<u>0.0953</u>
	HRA	0.0359	0.0513	0.0393	0.0223	0.4441	0.6462	0.0748	0.0952
NHiTS	Base	0.0516	0.0804	<u>0.0357</u>	0.0231	0.3040	0.3016	0.0725	0.0698
	BU	0.0517	0.0806	<u>0.0369</u>	0.0225	0.3396	0.3070	<u>0.0687</u>	0.0721
	RBU	<u>0.0510</u>	<u>0.0801</u>	0.0366	<u>0.0224</u>	0.3358	0.3052	0.0671	<u>0.0705</u>
	HRA	0.0503	0.0790	0.0353	0.0219	<u>0.3352</u>	<u>0.3040</u>	0.0666	0.0698
NLinear	Base	0.0504	0.0522	0.0518	<u>0.0369</u>	0.4492	0.4862	0.0820	0.0864
	BU	0.0489	0.0504	<u>0.0466</u>	0.0379	<u>0.4079</u>	<u>0.4462</u>	0.0800	0.0854
	RBU	<u>0.0482</u>	<u>0.0495</u>	0.0467	0.0381	0.4096	0.4474	<u>0.0797</u>	<u>0.0850</u>
	HRA	0.0467	0.0485	0.0444	0.0368	0.4057	0.4446	0.0792	0.0836
DeepAR	Base	0.0422	<u>0.0982</u>	0.0188	<u>0.0502</u>	0.4519	0.3304	0.1067	0.1119
	BU	<u>0.0409</u>	0.1037	<u>0.0151</u>	0.0515	<u>0.4499</u>	0.3174	0.1121	0.1102
	RBU	0.0415	0.1022	0.0152	0.0503	0.4676	0.3281	<u>0.1079</u>	<u>0.1101</u>
	HRA	0.0402	0.0980	0.0148	0.0483	0.4491	<u>0.3254</u>	0.1067	0.1095
Lag-llama	Base	<u>0.0561</u>	0.0769	<u>0.0239</u>	<u>0.0231</u>	0.3455	0.4225	0.0764	0.1279
	BU	0.0564	0.0760	0.0242	0.0242	0.3655	0.4438	0.0759	0.1266
	RBU	0.0562	<u>0.0753</u>	0.0242	0.0234	0.3661	0.4468	<u>0.0733</u>	<u>0.1255</u>
	HRA	0.0554	0.0752	0.0226	0.0224	<u>0.3632</u>	<u>0.4405</u>	0.0716	0.1211
Imp of HRA		1-5%		1-7%		0-4 %		1-4%	

decline in performance compared to the base forecast. Lag-Llama did not fine-tune effectively for the *Wiki* dataset, as it exhibited worse CRPS compared to the zero-shot prediction. HRA also demonstrates strong performance for longer forecast horizons, as evidenced by Table 3.

We also assess the performance of the probabilistic forecasting methods using CRPS. For illustrative purposes, we utilize DLinear as the base forecasting model and evaluate the performance of DLinear (Base), BU, RBU, and HRA across all four datasets. Weighted CRPS normalizes the error by dividing the errors by the actual data value, whereas CRPS is defined using actual errors. From Fig. 2, it is evident that HRA performs significantly better than the other SoTA models and reordering techniques. Detailed comparison using CRPS is included in Table3 and Table4 in the supplementary material.

4.2 Correlation Preservation

HRA preserves Spearman’s rank correlation between time series across all levels in the Bottom-Up Probabilistic Forecasting approach. To illustrate this claim, we consider our example in Fig. 1a of *Simulated* data and calculate the Spearman’s rank correlation matrix between all bottom-level series: $y_{AA}, y_{AB}, y_{BA}, y_{BB}$ from their historical values. For our benchmark dataset, we utilize the labour dataset, which comprises 57 time series, with 32 at the bottom level.

To demonstrate correlation preservation, we select the last 4 time series of the labour data, which depicts the employment status of Australia. We used DLinear model for base forecasting. Preserving correlation has applications in forecasting multivariate time series and multiple time series with high spatio-temporal covariance. The correlation matrix between all bottom level probabilistic forecasts is defined by sampling point time series from each probabilistic forecast and then averaging over all samples. In Fig. 3, we show heatmaps for Spearman’s rank correlation between bottom level series for historical values (Past Correlation), base forecast (Base), Revised Bottom-Up Forecast (RBU) and Heuristic Reordering Approach (HRA) for *Simulated* data and Fig. 4a and Fig. 4b shows comparison for *Labour* data. For Labour data, FFWA denotes *Employed full-time Females Western Australia*, and a similar pattern applies to the other series. It is evident that HRA outperforms the DLinear (Base), BU and RBU approaches even with $p = 0.1$. As the value of p increases, there should be better correlation preservation, as evidenced by Fig. 4a and Fig. 4b. There is a clear improvement in correlation for $p = 0.3$ compared to $p = 0.1$, and for $p \geq 0.3$, the correlation remains constant, indicating that almost all correlation is preserved for $p \geq 0.3$. We quantified the amount of correlation preserved by taking the mean of the absolute differences between past correlation matrices and the three correlation matrices (Base, RBU, HRA) defined on forecasted values. Details are presented in Table 4. Correlation plots for *Wiki* and *Tourism* datasets are included in supplementary.

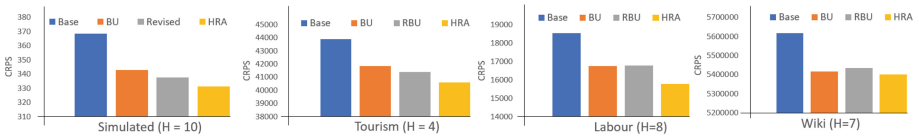
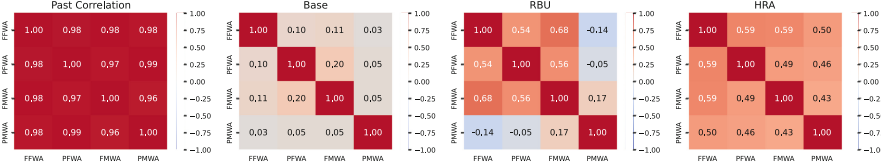


Fig. 2. Comparison of CRPS using DLinear as the base forecasting model. Lower loss values indicate better probabilistic forecasts. HRA consistently outperforms both the SoTA model and reordering models.



Fig. 3. Spearman’s rank correlation comparison for bottom level series of *Simulated* data with $H = 10$ and $p = 0.3$. HRA recovers past data correlation better than Base and RBU approaches.



(a) Spearman’s rank correlation comparison for bottom level series of *Labour* data for $p = 0.1$. HRA recovers past data correlation better than Base and RBU approaches.



(b) Spearman’s rank correlation comparison for bottom level series of *Labour* data for $p = 0.3$. It is evident that the higher the p value, the better the correlation preservation.



(c) Spearman’s rank correlation comparison for bottom level series of *Labour* data for $p = 0.5$. Correlation matrix for $p = 0.3$ and $p = 0.5$ are almost identical

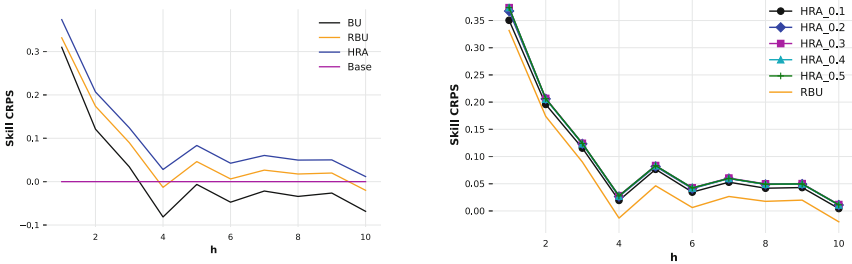
Fig. 4. Spearman’s rank correlation comparison for bottom-level series of *Labour* data with $H = 8$ at different p values. With the increase in p values, correlation increases, and for $p \geq 0.3$, the correlation matrix remains identical.

4.3 Effect of Extreme Values

We assessed the impact of the hyperparameter p on the performance of the HRA method, which involves reordering p fraction of Extreme Values from a sample set of size K . We saved the *Simulated* data comprising 310 data points, with 10 reserved for testing. Similarly, for the benchmark dataset, we utilized the

Table 4. Mean absolute correlation difference between past correlations with respect to correlation matrices defined using three methods (Base, RBU, HRA) with $p = 0.3$. A smaller difference indicates better preservation.

Data		Simulated	Labour	Wiki	Tourism
Mean Absolute Correlation Difference	Base	0.705	0.6625	0.2302	0.1789
	RBU	0.6037	0.5125	0.3608	0.2423
	HRA	0.2375	0.2925	0.2079	0.1689



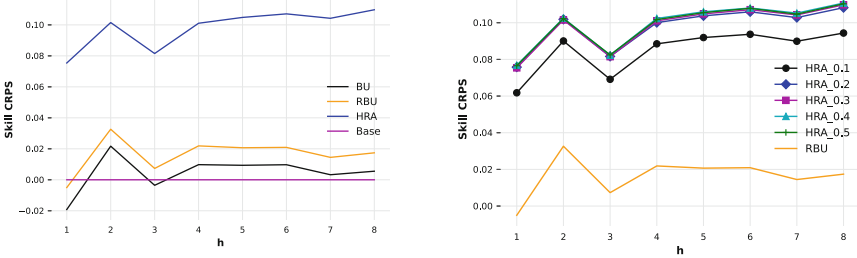
(a) Comparing Skill-CRPS for all four methods. Skill CRPS for HRA is computed with $p = 0.1$. (b) Effect of different values of p on Skill-CRPS of HRA

Fig. 5. Effect of Extreme Values on Skill-CRPS for *Simulated* data. We took fraction $p = [0.1, 0.2, 0.3, 0.4, 0.5]$ and measured the corresponding effect on Skill-CRPS for the HRA. HRA outperforms all three methods even with just $p = 0.1$. For $p \geq 0.3$, the Skill CRPS remains constant.

Labour data, setting aside 8 data points for testing. Base probabilistic forecasts are generated using DLinear. We took $p = [0.1, 0.2, 0.3, 0.4, 0.5]$ to assess its impact on enhancing the Skill-CRPS. Suppose $p = 0.1$, then we take 10% as the Extreme Values from the total number of samples and reorder them using HRA. Fig. 5a shows the Skill-CRPS comparison with $p = 0.1$ for DLinear (Base), BU, RBU and HRA. Fig. 5b shows the change in Skill-CRPS with a change in values of p . The orange line shows the Skill-CRPS for RBU, and others are of HRA for different values of p . Similar comparisons are depicted for the *Labour* data in Figures 6a and 6b. As the value of p increases, we observe an improvement in performance. For $p = 0.3$, we get the best Skill-CRPS; for $p = 0.4, 0.5$ Skill-CRPS converges to the same value (hence overlapped in the plot). It is evident that HRA outperforms all other methods, and a value of $p = 0.1$ is sufficient to surpass the performance of RBU.

5 Conclusion

In summary, we propose an Extreme Value reordering approach to generate coherent hierarchical probabilistic forecasts. The core idea behind our approach



(a) Comparing Skill-CRPS for all four methods. Skill CRPS for HRA is computed with $p = 0.1$. (b) Effect of Extreme Values on Skill-CRPS of HRA

Fig. 6. Effect of Extreme Values on Skill-CRPS for *Labour* data. We took fraction $p = [0.1, 0.2, 0.3, 0.4, 0.5]$ and measured the corresponding effect on Skill-CRPS for the HRA. HRA outperforms all three methods even with just $p = 0.1$. For $p \geq 0.3$, the Skill CRPS remains constant.

is that by reordering the Extreme Values within predictive samples to minimize the objective defined on correlations, we can significantly enhance the quality of the probabilistic distribution for time series data. Even if the accuracy improvements are not substantial, the preservation of correlation structures plays a crucial role in ensuring that the forecasts remain reliable and aligned with the inherent dependencies in the data. Notably, our HRA demonstrated significant correlation preservation compared to existing methods. What makes our method unique is that it can improve forecasts by preserving the correlation between the time series, which is not considered in the forecasting phase. Compared to SoTA models, our HRA model effectively captures correlations and improves Skill-CRPS across simulated and well-established benchmark datasets. HRA also performed well for the longer forecast horizon, exhibiting a performance improvement of up to 7% compared to the existing SoTA. However, it is important to note a potential drawback of our method: there may be situations where the improved Skill-CRPS may not converge for small values of fraction of Extreme Values p , and the computational time may increase. However, in all our experiments, we have observed that convergence typically occurs at a small p value, making our approach a practical and effective solution for many forecasting scenarios.

References

1. Australian Bureau of Statistics (2023), <https://www.abs.gov.au/statistics/labour>
2. Athanasopoulos, G., Hyndman, R.J.: Modelling and forecasting australian domestic tourism. *Tour. Manage.* **29**(1), 19–31 (2008)
3. Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: N-HiTS: Neural hierarchical interpolation for time series forecast-

- ing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 6989–6997 (2023)
4. Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **102**(477), 359–378 (2007)
 5. Hyndman, R.J., Athanasopoulos, G.: *Forecasting: principles and practice*. OTexts (2018)
 6. Kamarthi, H., Kong, L., Rodríguez, A., Zhang, C., Prakash, B.A.: Proffit: Probabilistic robust forecasting for hierarchical time-series. arXiv preprint [arXiv:2206.07940](https://arxiv.org/abs/2206.07940) (2022)
 7. Palaskar, S., Ekambaram, V., Jati, A., Gantayat, N., Saha, A., Nagar, S., Nguyen, N.H., Dayama, P., Sindhgatta, R., Mohapatra, P., et al.: Automixer for improved multivariate time-series forecasting on bizitops data. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 22962–22968 (2024)
 8. Rasul, K., Ashok, A., Williams, A.R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N.V., Schneider, A., et al.: Lag-llama: Towards foundation models for time series forecasting. arXiv preprint [arXiv:2310.08278](https://arxiv.org/abs/2310.08278) (2023)
 9. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **36**(3), 1181–1191 (2020)
 10. Taieb, S.B., Taylor, J.W., Hyndman, R.J.: Coherent probabilistic forecasts for hierarchical time series. In: International Conference on Machine Learning. pp. 3348–3357. PMLR (2017)
 11. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 11121–11128 (2023)



TS-NUC : Nearest Unlike Cluster Guided Generative Counterfactual Estimation for Time Series Classification

Ayanabha Ghosh¹(✉), Rishi Jain², Shubham Parida³, and Debasis Das¹

¹ Indian Institute of Technology Jodhpur, Jodhpur, India
p23iot002@iitj.ac.in , debasis@iitj.ac.in

² Rajiv Gandhi Institute of Petroleum Technology, Amethi, India
22it3038@rgipt.ac.in

³ Techno India University, Kolkata, India

Abstract. Machine learning is a cornerstone of modern decision-making systems, yet its inner workings often remain a mystery to human stakeholders. Bridging this gap requires clear, human-understandable explanations of how these models transform inputs into outputs. One effective approach for achieving this type of transparency is through counterfactual explanations. Counterfactuals inform users about what changes need to be made and why, offering recommendations on how to alter an undesired outcome into a desired one, which ultimately enhances the comprehensibility and reliability of machine learning models. In this work, we propose TS-NUC, a novel model-agnostic counterfactual generation approach dedicated to the domain of time series classification. Our approach consists of a pre-trained LSTM-Autoencoder which generates the latent representation of a time series instance. By optimizing the latent representation, guided by the user-provided target class latent cluster, TS-NUC is capable of generating high-quality counterfactual explanation. Through extensive experiments on a total of 5 datasets from the UCR archive and performance comparison with latest state-of-the-art approaches on three popularly used evaluation metrics, namely Validity, Proximity and Compactness, we show that our approach produces comparable and better results.

Keywords: Counterfactual explanations · Time series classification · Deep learning · LSTM Autoencoder · Explainable AI

1 Introduction

Counterfactual explanation of an instance deals with generating or proposing modifications in feature values in order to classify it into the desired class from an undesired class so as to make the classifier explainable and bring transparency in its decision making process. Generating counterfactual explanations has a major

R. Jain and S. Parida—Authors have equal contributions.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15326, pp. 392–406, 2025.
https://doi.org/10.1007/978-3-031-78395-1_26

role in understanding the contributions of features on a specific prediction done by the model. Also, in various real life applications, counterfactuals help end users get insights about a certain outcome by giving justification(s) of their inputs with the help of another possible input which might have led to the desired outcome expected by the user.

Most of the state-of-the-art methods have been proposed in generating counterfactual explanations of image data having spatial dependency among the features through black-box or white-box approach [1, 22, 26]. However, generating counterfactual explanations of time-series data is more complex due to the presence of temporal dependency among the sequential instances. For example, Electrocardiogram (ECG) signal classification, sensor signal classification, and stream monitoring data are sequential in nature, and therefore, it is common to model such data as sequence of events in order to efficiently address the prediction and/or classification task. In recent times, autonomous vehicles and electric vehicles have used temporal models for predicting failures and the probable time to undergo maintenance. Providing an efficient counterfactual-based explainable approach will help the users understand which part(s) of their vehicle and/or what other conditions are indicating potential failure(s) and/or required maintenance of the vehicle[23]. In AI-driven medical diagnosis, counterfactuals can provide example-specific explanations by which stakeholders can infer the necessary modifications to change the prediction from an undesired state to a desired state[32]. Most of such sequential machine learning models are black-boxes, preventing understandability of their internal functionality and usage in the application areas where transparency and explainability are a primary concern for trust and reliability[32].

To address such challenges, we propose a novel model-agnostic methodology on generating counterfactual explanations for univariate time series classification tasks. Given a time series instance and a trained black-box classifier model, our approach aims to generate its counterfactual explanation which, by definition, the classifier will classify into a specific user-given target class by incorporating minimum changes in the feature values. For this, an LSTM-based autoencoder[13] has been used which will capture the representations of the data in hand. By optimizing the latent space representation, the CF generator can be forced to produce a counterfactual of a specific example belonging to a specific class. Extensive experiments using multiple publicly available and benchmark time-series classification datasets from UCR Archive[6] and comparisons with different state-of-the-art works show that our proposed approach can produce around 3% - 5% improved counterfactual explanations in terms of Validity, Proximity and Compactness. Through this work, our main contributions are as follows :

1. We have proposed a LSTM-based Autoencoder for generating the counterfactual sample of a given univariate time-series instance. By optimizing the latent representation of a time-series instance, we can obtain the counterfactual of that instance.

2. We have optimized a custom objective function in order to change the latent representation, which, when passed through LSTM-decoder, produces the necessary counterfactual sample. We employ a weighted loss function which essentially maintains a trade-off between the individual components of it.
3. The optimization has been constrained by Nearest Unlike Cluster Centroid (NUC), which distracts the optimizer and forces the latent representation to move towards a given NUC. Using this approach, given a time series instance from a class, counterfactual generation can be controlled by providing a specific target class label, which the user wants to generate a counterfactual of.

The further article has been organized as mentioned - §2 discusses some of the recent and closely related works on generating counterfactuals for time series classification, §3 discusses the problem definition and the relevant notations we have used throughout the article, which is followed by detailed explanation of our proposed methodology, datasets and training setup in §4. Finally, the results obtained have been analyzed in §5 and the future research scopes in §6.

2 Related Works

In recent years, several machine learning techniques have been used to propose a diverse set of time series classification algorithms[2]. We have categorized such closely related works into three parts: Conventional approaches in §2.1, Shapelet based approaches in §2.2 and Feature importance based approaches in §2.3. Moreover, some other approaches not falling under these above categories have been kept in the last subsection §2.4.

2.1 Conventional Approaches

Symbolic aggregate approximation (SAX)[12] algorithm bins continuous time series into intervals, transforming independently each time series (a sequence of floats) into a sequence of symbols, usually letters or strings. Although, SAX has been recently employed in the development of interpretable time series classifiers, such as XEM[9] and PETSC[10], due to its inherent characteristics, it has some limitations. The discretization of time series signal can lead to a significant loss of information, especially if the original time series contains subtle but important variations that are not captured by the symbolic representation. Moreover, SAX relies on the assumption that the normalized time series data follows a Gaussian distribution to determine breakpoints for symbolic conversion. If the data does not follow a Gaussian distribution, the symbolic representation might not be accurate.

Shapelet-based approaches utilize time series subsequences, known as shapelets, as discriminative features for training classifiers, such as random forests and SVMs[2,17]. In addition, HIVE-COTE[19] was introduced as an

ensemble technique that combines various classifiers such as elastic ensembles and shapelet transform, with a hierarchical voting structure, surpassing all prior approaches in performance. Karlsson et al.[18] put out a method that introduces disturbances to time series data either in a localized or global manner, led by either the random shapelet forest classifier or the k-nearest neighbor classifier, respectively. This approach offers model-specific explanations, therefore, it is not applicable to any other classifier. Shapelet based approaches, though promising for their interpretability and effectiveness in time series classification, face several challenges and limitations. Shapelet discovery and matching can be computationally intensive, especially with large datasets and high-dimensional time series. The search for optimal shapelets requires substantial computational resources, which can be a bottleneck in real-time applications. Shapelet-based methods may not always be robust to noise and variability in the data. Small perturbations or variations in the time series can lead to significant changes in the identified shapelets, affecting the stability and reliability of the counterfactual explanations.

2.2 Latent Representation based Approaches

In the TSC domain, researchers have recently explored the use of latent representations to generate explainable findings. The counterfactual generation techniques proposed by Pawelczyk et al.[24], Joshi et al.[15], Balasubramanian et al.[3], and Van Looveren & Klaise [29] involve learning latent representations of each class through the use of an Autoencoder (AE) or a Variational Autoencoder (VAE). However, their primary emphasis is on tabular or picture data, and none of them have been utilized for TSC. The LASTS[11] method was developed using autoencoders to create factual and counterfactual rules. These rules are generated by training a local latent decision tree classifier on the original time series data. The rules specify that the original time series must include or exclude particular shapelets in order to achieve the desired or original class. In their 2021 study, Wang et al.[31] introduced LatentCF++ as a method for acquiring latent space representations for time series counterfactuals.

2.3 Feature Importance based Approaches

Alternative approaches to explainable machine learning have been suggested to elucidate model predictions through the provision of feature importance scores. For instance, one method involves generating local model-agnostic explanations (LIME) by randomly altering input samples to fit surrogate models[25]. Another approach involves computing Shapely values to estimate the significance of features for a given classifier[20]. Sivill et al.[27] introduced LIMESegment as a method to adapt these techniques for the time series domain. They utilized a nearest neighbour-based approach combined with harmonic analysis to create reliable perturbations for the surrogate model. This allowed them to extract temporal segments of time series along with their corresponding local importance scores. In contrast, Bento et al.[4] introduced TimeSHAP, a method that offers

relevance scores at the feature-level, timestep-level, and cell-level for recurrent neural network models. As far as we know, none of these methods for determining the importance of features has been taken into account or included in the production of counterfactuals for time series data. Furthermore, LIME and SHAP based methods treat each time point independently, potentially ignoring temporal dependencies, which may lead to explanations that fail to capture the sequential nature of the data. Also, LIME’s local approximations can become less reliable in such high-dimensional spaces, making the explanations harder to interpret. In case of SHAP, the computational cost increases significantly with the number of features, making it less practical for long time series or very high-dimensional datasets.

2.4 Other Approaches

Researchers have introduced various advanced TSC algorithms that utilize deep learning techniques. These include LSTM-FCN[16], InceptionTime[14], and convolutional feature transforms like ROCKET[8]. These algorithms have shown similar benchmark metrics while also improving the scalability of the models. However, because of their opaque nature, they lack the ability to be understood and explained[5]. Delaney et al.[7] introduced the Native Guide approach for classifying time series using CNNs. This method incorporates CAM feature weight vectors to systematically modify a portion of the time series in order to produce counterfactual explanations. Similar to [18], this approach also offer model-specific explanations, thus, not applicable to any other classifier.

3 Problem Definition and Notation

Let X be a time-series instance with \mathbf{t} timestep records i.e. $X = \{x_1, \dots, x_t\}$ where each $x_i \in \mathbb{R}^d$ and $d \in \mathbb{Z}^+$ and the corresponding label be y , where $y \in \mathcal{Y}$ and \mathcal{Y} is the set of all possible class labels. From the definition, the counterfactual of X will be a modified version of it, say X_{cf} , such that the blackbox classifier $\mathcal{F}_{CLF}(\cdot)$ classifies it to a specific target class y_{cf} . Provided the total number of classes is K and the raw output of the classifier as $Z = [z_1, z_2, \dots, z_K]$, then the final decision of $\mathcal{F}_{CLF}(\cdot)$ can be given as,

$$\hat{y} = \underset{p}{\operatorname{argmax}} p_i \quad \text{where } p_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \forall z_i \in Z$$

Now, we additionally define two models, the first one is a simple unidirectional LSTM-based Autoencoder (denoted as LSTM-AE in the rest of the article) which can capture the temporal nature of the data and reconstruct the input time series instance by maintaining the same. The second one is a Multilayer perceptron or ANN based Autoencoder (denoted as MLP-AE in the rest of the article) which is capable of reconstructing the hidden representation of a time series instance. By perturbing and optimizing the MLP-AE, we aim to generate a

modified representation of the hidden vector, generated by the encoder of LSTM-AE, which the decoder of LSTM-AE takes as input and reconstruct a modified instance, so that the black box classifier (denoted as LSTM-CLF in the rest of the article) can classify it to another target class. The custom optimization objective is designed such that it takes into account the user-provided target class and generate a counterfactual from the same.

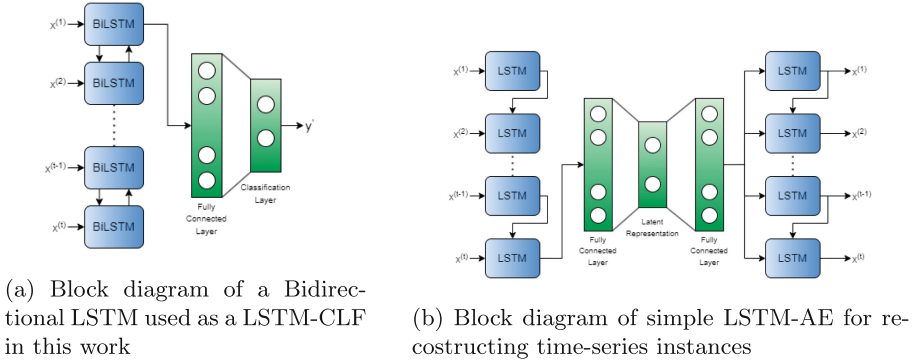


Fig. 1. Block diagrams of LSTM-CLF and LSTM-AE

Table 1. Hyperparameter and Training Setup of individual modules; H-Dim denotes the dimension of hidden layer, Enc. size denotes the size of the encoded vector and Val. Acc. is Validation Accuracy

Model	Nature	#Layers	H-Dim	Enc. size	Val. Acc.
LSTM-CLF	Bi-LSTM	8	128	—	98%
LSTM-AE	LSTM	2	256	128	—
MLP-AE	ANN	3	128	128	—

4 Proposed TS-NUC Framework

In this section, we describe our proposed counterfactual generation approach TS-NUC. The different modules and components have been discussed in the subsequent subsections within this section.

4.1 Black-box Classifier

Our proposed approach is model-agnostic, i.e. it does not use any information related to the black-box classifier, such as weights, layers, and so on and we only have access to the predict function of the model. In this experiment, we have used a Bidirectional LSTM model trained on individual datasets separately. The architecture block diagram and hyperparameter setup have been shown in Fig.1a and Table-1 respectively. The loss function used in training is Crossentropy loss.

4.2 LSTM-Autoencoder

The LSTM-Autoencoder consists of two single-layer unidirectional LSTM, one in Encoder and one in decoder, which learns to reconstruct the time-series instances. In addition, we have implemented it to exploit the Encoder and Decoder modules separately during counterfactual generation whose details have been discussed later. By optimizing the latent representation of a time-series instance, we aim to generate the counterfactual of the same. The architecture block diagram and hyperparameter setup have been shown in Fig.1b and Table-1 respectively. The loss function used in training is Mean Squared Error.

4.3 ANN based Autoencoder

We first construct a multilayer perceptron based autoencoder (MLP-AE) which learns to reconstruct the latent representations of the time series instances of an entire training dataset. We use the pre-trained LSTM-AE for generating the latent representation of an instance, forward pass it through MLP-AE and train to reconstruct it. For this experiment, the hyperparameter setup has been mentioned in Table-1. Similar to LSTM-AE, the loss function used here is Mean Squared Loss. Additionally, we have incorporated Dropout regularization with a probability of 0.25 after first layer to prevent the model from overfitting.

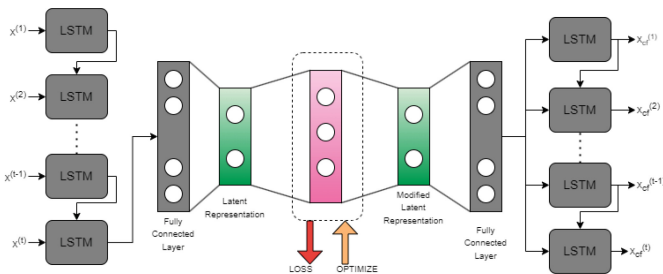


Fig. 2. Block diagram of the proposed TS-NUC framework. The gray coloured boxes indicate pre-trained weights of the respective modules of LSTM-AE

4.4 Computation of Centroid

To facilitate the counterfactual generation algorithm push the MLP-AE towards generating a latent representation of user given target class, we compute the centroids of each class clusters from the corresponding latent vectors belonging to a specific class. Thus, for K number of classes, we compute K centroids for each set of latent vectors belonging to a specific class. For a given class label c and the corresponding set of latent vectors generated from the set of instances X_c belonging to class c be Z_c , the centroid μ_c is defined as the mean of all the latent vectors in Z_c which can be denoted as,

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} z_i, \forall z_i \in Z_c$$

where $N_c = |Z_c|$, the number of latent vectors belonging to class c .

The counterfactual generation approach utilizes the pre-computed centroid of the user-given target class to optimize and force the MLP-AE to generate a desired latent representation which the decoder of LSTM-AE can create a sample from the target class and LSTM-CLF can classify it to the same target class.

4.5 Counterfactual generation

After training of all the modules, the main idea is to perturb and optimize the MLP-AE in order to generate a modified latent representation so that the decoder can reconstruct a modified sample and the black box classifier will classify the new sample into the desired target class. The block diagram and the algorithm of the same have been shown in Fig.2 and algorithm-1 respectively.

At first, the algorithm takes a single time-series instance X , a desired target class y_{cf} and the corresponding pre-computed centroid $\mu_{y_{cf}}$. Then it optimizes a loaded instance of pre-trained MLP-AE using a customized objective function (discussed in §4.6). Additionally, to prevent the optimization process being stuck at a certain minima, at every iteration, we add Gaussian noise sampled from $\mathcal{N}(-1, 1)$ with the generated latent vector. Experimentally observed, it has reduced the number of steps performed by the algorithm and resulted in faster convergence. The maximum number of iterations has been set to 10^5 .

4.6 Objective Function

As already mentioned earlier, we have used a custom weighted objective function to optimize the pre-trained MLP-AE. It consists of three different components. First, Crossentropy loss is computed from the LSTM-CLF module between the target class y_{cf} and the predicted class \hat{y} . Second, the Mean Average Error is computed between the actual input X and the output of the LSTM-AE decoder i.e. the modified sample \hat{X} . Third, the euclidean distance between the output of the optimized MLP-AE i.e. the modified latent vector \hat{z} and the pre-computed

Algorithm 1 Proposed TS-NUC Algorithm

Require: Trained Blackbox model $\mathcal{F}_{CLF}(\cdot)$, pre-trained LSTM-Autoencoder $\mathcal{F}_{AE}(\cdot)$ consisting of Encoder module $\mathcal{F}_{AE}^E(\cdot)$ and Decoder module $\mathcal{F}_{AE}^D(\cdot)$, pre-trained MLP Autoencoder $\mathcal{F}_Z(\cdot)$, weighted counterfactual Loss function $\mathcal{L}(\cdot)$

Input: $X \leftarrow \{x_1, \dots, x_t\}$, corresponding ground truth y , target class y_{cf} and corresponding pre-computed class centroid $\mu_{y_{cf}}$

Output: Generated counterfactual \hat{X}

- 1: Initialize iter $\leftarrow 0$, max_iter $\leftarrow 10^4$, $\alpha \leftarrow 10^{-3}$
- 2: $z \leftarrow \mathcal{F}_{AE}^E(X)$
- 3: $\hat{z} \leftarrow \mathcal{F}_Z(z) + \mathcal{N}(-1, 1)$
- 4: $\hat{X} \leftarrow \mathcal{F}_{AE}^D(\hat{z})$
- 5: $\hat{y} \leftarrow \mathcal{F}_{CLF}(\hat{X})$
- 6: **while** $\hat{y} \neq y_{cf} \wedge \text{iter} < \text{max_iter}$ **do** ▷ Counterfactual generation loop
- 7: $L \leftarrow \mathcal{L}(X, \hat{X}, y_{cf}, \hat{y}, \hat{z}, \mu_{y_{cf}})$
- 8: Optimize \mathcal{F}_Z ▷ Uses Adam optimizer with learning rate α
- 9: $\hat{z} \leftarrow \mathcal{F}_Z(z) + \mathcal{N}(-1, 1)$ ▷ Generating optimized latent vector \hat{z}
- 10: $\hat{X} \leftarrow \mathcal{F}_{AE}^D(\hat{z})$
- 11: $\hat{y} \leftarrow \mathcal{F}_{CLF}(\hat{X})$ ▷ Prediction of new \hat{X}
- 12: $z \leftarrow \hat{z}$
- 13: **end while**
- 14: **return** $\hat{X} = \{x_1^{cf}, \dots, x_t^{cf}\}$

centroid of y_{cf} , i.e. $z_{y_{cf}}$. The objective function can be mathematically represented as,

$$\mathcal{L}(\cdot) = -\lambda_1 \cdot \sum_{i=1}^K y_{cf_i} \log(\hat{y}_i) + \lambda_2 \cdot \frac{1}{T} \sum_{i=1}^T \left| X_i - \hat{X}_i \right| + \left[\sum_{i=1}^D (\hat{z}_i - \mu_{y_{cf_i}})^2 \right]^{\frac{1}{2}}$$

where K is the total number of classes, T is the number of time steps in a single instance and D is the dimensionality of latent representation vector in \mathcal{F}_{AE} , λ_1 and λ_2 are coefficients which have been experimentally set as 0.05 and 0.01 respectively.

4.7 Implementation

The whole system has been implemented in PyTorch v2.2.0 with support of CUDA v12.1 and trained on a single NVIDIA RTX 3060 GPU with 6GB VRAM. For training all three components, i.e. LSTM-CLF, LSTM-AE and MLP-AE models, we have used Adam Optimizer with β_1 and β_2 values of 0.9 and 0.999 respectively. The maximum number of epochs is 10^5 and the initial learning rate has been set as 10^{-3} which is being reduced by a factor of 0.1 if the validation loss does not decrease after 25 consecutive epochs. Additionally, we have incorporated Early stopping regularization based on the validation loss, which stops the training if the validation loss does not decrease for 100 consecutive epochs.

5 Evaluation Metrics and Results

For evaluating our proposed approach, we have chosen three popularly used evaluation metrics namely Validity, Proximity and Compactness. Moreover, we have chosen five closely related works, which have been proposed on time-series counterfactual generation, for comparing our proposed TS-NUC approach. This section discusses about each of the metrics, the obtained results and comparison with the chosen state-of-the-art methods.

5.1 Validity

Validity[21,28] is defined as whether the generated sample is being classified in the target class or not. That means, if the generated sample is classified in the target class, it is considered to be valid counterfactual, else, not a valid counterfactual. Mathematically, it can be expressed as,

$$\text{Validity}(\mathcal{X}_{cf}) = \frac{\#(\hat{y} = y_{cf})}{N}$$

where \mathcal{X}_{cf} is the set of generated counterfactuals, \hat{y} is predicted class label of a single instance and y_{cf} is the target class. It is to be noted that this metric has been computed for the entire dataset. For a single instance, the validity yields binary output i.e. either 1 or 0. The higher the validity, the better the performance of the model.

Table-2 shows the quantitative comparison on validity with the chosen state-of-the-art approaches. As we can see, for the ItalyPower, FordB and MoteS-train datasets, TS-NUC shows the best results as compared to the other methodologies. For the FordA and TwoLeadECG datasets, LatentCF++[31] and Glacier[30] yield better performance over TS-NUC in terms of validity.

Table 2. Comparison of **Validity**↑ with other State-of-the-art methods

Methods	ItalyPower	FordA	FordB	TwoLeadECG	MoteStrain
LatentCF++ [31]	0.9263	0.9344	0.9675	0.9052	0.9618
Glacier [30]	0.8205	0.9320	0.8840	0.984	0.9585
FGD [18]	0.4523	0.6535	0.6251	0.5298	0.6795
KNN [18]	0.7926	0.8903	0.7852	0.7452	0.8847
RDF [17]	0.8544	0.8523	0.8422	0.8263	0.9052
TS-NUC (Ours)	0.9510	0.9025	0.9754	0.9685	0.9824

5.2 Proximity

Proximity[21] is the measure of how close the generated counterfactual is to the original input time-series instance. It is defined as the feature-wise distance between the generated counterfactual and the input. In our experiment, we have chosen the distance measure to be simple Euclidean distance averaged over the total number of time steps. Mathematically, it can be expressed as,

$$\text{Proximity}(X, X^{cf}) = \frac{1}{T} \left[\sum_{i=1}^T (X_i - X_i^{cf})^2 \right]^{\frac{1}{2}}$$

where X is the original sample, X^{cf} is the generated counterfactual and T is the number of time steps. Proximity can be measured for a single instance. The lower the proximity, the better the model performance.

Table-3 shows the quantitative comparison with other chosen state-of-the-art methods and chosen datasets. Here, we have computed proximity for all the samples in Test set and taken the mean of it. As we can see, TS-NUC shows best performance among all methodologies for ItalyPower, FordA and TwoLeadECG datasets. Counterfactuals generated for FordB dataset using Glacier[30] are the closest to the original samples. Similarly, counterfactuals of MoteStrain dataset generated using RSF[17] shows lowest proximity.

Table 3. Comparison of **Proximity**↓ with other State-of-the-art methods

Methods	ItalyPower	FordA	FordB	TwoLeadECG	MoteStrain
LatentCF++ [31]	0.4785	0.4368	0.6099	0.1839	0.3884
Glacier [30]	0.218	0.345	0.098	0.189	0.475
FGD [18]	0.3373	0.2387	0.2176	0.2655	0.4798
KNN [18]	0.3633	2.0811	2.1105	0.1936	1.095
RDF [17]	0.2513	0.4820	0.3934	0.1793	0.1903
TS-NUC (Ours)	0.1952	0.1544	0.1454	0.1652	0.2065

5.3 Compactness

Compactness is defined as the mean of the feature-wise absolute differences between the generated counterfactual and the original sample. It captures the overall change in the feature values from original sample to the counterfactual of it. A similar formulation has been given in Glacier[30], however, the authors designed it for the entire dataset and not for a single instance. In contrast, we have defined it for a single instance. It can be expressed as follows,

$$\text{Compactness}(X, X^{cf}) = \frac{1}{T} \cdot \sum_{i=1}^T |X_i - X_i^{cf}|$$

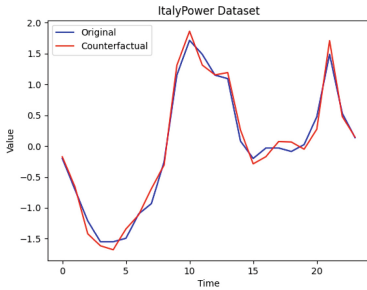
where X is the original sample, X^{cf} is the generated counterfactual and T is the number of time steps. The lower the compactness, the better the model performance.

Table-4 shows the quantitative comparison of the chosen methods on compactness. For evaluation purposes, we have computed the average of the Compactness on all the samples on validation split. It clearly shows that our proposed TS-NUC shows best results for ItalyPower, FordB and MoteStrain datasets. For the rest of the two datasets, Glacier[30] shows the best performance, though, TS-NUC is quite close to Glacier for these two datasets.

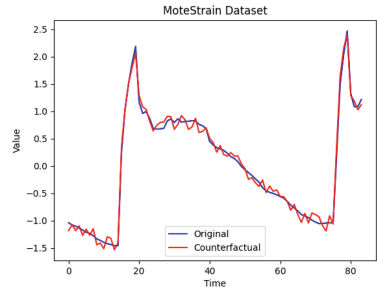
Table 4. Comparison of **Compactness**↓ with other State-of-the-art methods

Methods	ItalyPower	FordA	FordB	TwoLeadECG	MoteStrain
LatentCF++ [31]	0.2543	0.1534	0.1854	0.1545	0.1487
Glacier [30]	0.2567	0.0982	0.1635	0.1254	0.1363
FGD [18]	0.4532	0.1956	0.2398	0.2301	0.2543
KNN [18]	0.3176	0.117	0.3128	0.1984	0.2388
RDF [17]	0.4549	0.1367	0.2568	0.1786	0.2785
TS-NUC (Ours)	0.1665	0.1076	0.1564	0.1387	0.1049

Fig.3 shows a sample output of counterfactual samples generated using TS-NUC of two randomly chosen samples from ItalyPower dataset (Fig.3a) and MoteStrain dataset (Fig.3b). As seen in Fig.3a, the counterfactual explanation shows some stochasticity, which we leave for future works.



(a) ItalyPower dataset



(b) MoteStrain dataset

Fig. 3. Sample input and corresponding counterfactual output from TS-NUC. The original input is shown in Blue curve while the corresponding generated counterfactual by the proposed TS-NUC is shown in Red curve.

6 Conclusion

In this work, we propose TS-NUC, a novel model-agnostic time-series classification counterfactual generation approach. Unlike previously proposed approaches, TS-NUC captures the temporal nature of the data by applying LSTM-based architectures instead of CNN or ordinary ANN based Autoencoders. It utilizes the latent representation of an input time-series instance and by perturbing and optimizing that, it generates a modified latent representation. The decoder module generated a modified instance from the new latent representation and the black-box classifier is expected to classify it into a desired class provided by the user. Experiments on different benchmark datasets from UCR Time Series Archive [6] and comparisons with different closely related state-of-the-art approaches show the effectiveness of our proposed approach on three different metrics namely Validity, Proximity and Compactness.

As future work, we plan to extend this work for different data modalities having temporal dependencies such as video, audio, text and so on. Moreover, we plan to modify and apply this method to specific application areas such as predictive maintenance, medical diagnosis, human activity recognition, which will enable the black-box decision makers explain their decisions on the basis of choosing features and their importance. The perspective of causal counterfactual generation for time series data can also be explored in future works.

Acknowledgement. This work is partially supported by the DST IIT Bhilai Innovation and Technology Foundation (IBITF) (IBITF/Note/EIR-PRAYAS/Cohort-03/SanctionLetter/2024-25/0076).

References

1. 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), 1–46 (07 2015). <https://doi.org/10.1371/journal.pone.0130140>
2. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Disc.* **31**, 606–660 (2017)
3. Balasubramanian, R., Sharpe, S., Barr, B., Wittenbach, J., Bruss, C.B.: Latentcf: a simple baseline for reverse counterfactual explanations. *arXiv preprint arXiv:2012.09301* (2020)
4. Bento, J., Saleiro, P., Cruz, A.F., Figueiredo, M.A., Bizarro, P.: Timeshap: Explaining recurrent models through sequence perturbations. In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. pp. 2565–2573 (2021)
5. Christoph, M.: *Interpretable machine learning: A guide for making black box models explainable*. Leanpub (2020)
6. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica* **6**(6), 1293–1305 (2019)

7. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. *CoRR abs/2009.13211* (2020), <https://arxiv.org/abs/2009.13211>
8. Dempster, A., Petitjean, F., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Disc.* **34**(5), 1454–1495 (2020)
9. Fauvel, K., Fromont, É., Masson, V., Faverdin, P., Termier, A.: Xem: An explainable-by-design ensemble method for multivariate time series classification. *Data Min. Knowl. Disc.* **36**(3), 917–957 (2022)
10. Feremans, L., Cule, B., Goethals, B.: Petsc: pattern-based embedding for time series classification. *Data Min. Knowl. Disc.* **36**(3), 1015–1061 (2022)
11. Guidotti, R., Monreale, A., Spinnato, F., Pedreschi, D., Giannotti, F.: Explaining any time series classifier. In: 2020 IEEE second international conference on cognitive machine intelligence (CogMI). pp. 167–176. IEEE (2020)
12. He, Z., Long, S., Ma, X., Zhao, H.: A boundary distance-based symbolic aggregate approximation method for time series data. *Algorithms* **13**(11), 284 (2020)
13. Homayouni, H., Ghosh, S., Ray, I., Gondalia, S., Duggan, J., Kahn, M.G.: An autocorrelation-based lstm-autoencoder for anomaly detection on time-series data. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 5068–5077 (2020). <https://doi.org/10.1109/BigData50022.2020.9378192>
14. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Disc.* **34**(6), 1936–1962 (2020)
15. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint [arXiv:1907.09615](https://arxiv.org/abs/1907.09615) (2019)
16. Karim, F., Majumdar, S., Darabi, H., Chen, S.: Lstm fully convolutional networks for time series classification. *IEEE access* **6**, 1662–1669 (2017)
17. Karlsson, I., Papapetrou, P., Boström, H.: Generalized random shapelet forests. *Data Min. Knowl. Disc.* **30**(5), 1053–1085 (2016). <https://doi.org/10.1007/s10618-016-0473-y>
18. Karlsson, I., Rebane, J., Papapetrou, P., Gionis, A.: Locally and globally explainable time series tweaking. *Knowl. Inf. Syst.* **62**(5), 1671–1700 (2020)
19. Lines, J., Taylor, S., Bagnall, A.: Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In: 2016 IEEE 16th international conference on data mining (ICDM). pp. 1041–1046. IEEE (2016)
20. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. *Advances in neural information processing systems* **30** (2017)
21. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. p. 607–617. FAT* '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3351095.3372850>, <https://doi.org/10.1145/3351095.3372850>
22. Oh, S.J., Schiele, B., Fritz, M.: Towards Reverse-Engineering Black-Box Neural Networks, pp. 121–144. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6_7, https://doi.org/10.1007/978-3-030-28954-6_7
23. Pashami, S., Nowaczyk, S., Fan, Y., Jakubowski, J., Paiva, N., Davari, N., Bobek, S., Jamshidi, S., Sarmadi, H., Alabdallah, A., et al.: Explainable predictive maintenance. arXiv preprint [arXiv:2306.05120](https://arxiv.org/abs/2306.05120) (2023)

24. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: Proceedings of the web conference 2020. pp. 3126–3132 (2020)
25. 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)
26. 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: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 618–626 (2017). <https://doi.org/10.1109/ICCV.2017.74>
27. Sivill, T., Flach, P.: Limesegment: Meaningful, realistic time series explanations. In: International Conference on Artificial Intelligence and Statistics. pp. 3418–3433. PMLR (2022)
28. Stepin, I., Alonso, J.M., Catala, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access* **9**, 11974–12001 (2021). <https://doi.org/10.1109/ACCESS.2021.3051315>
29. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 650–665. Springer (2021)
30. Wang, Z., Samsten, I., Miliou, I., Mochaourab, R., Papapetrou, P.: Glacier: guided locally constrained counterfactual explanations for time series classification. *Machine Learning* pp. 1–31 (2024)
31. Wang, Z., Samsten, I., Mochaourab, R., Papapetrou, P.: Learning time series counterfactuals via latent space representations. In: Soares, C., Torgo, L. (eds.) *Discovery Science*, pp. 369–384. Springer International Publishing, Cham (2021)
32. Wang, Z., Samsten, I., Papapetrou, P.: Counterfactual explanations for survival prediction of cardiovascular icu patients. In: Tucker, A., Henriques Abreu, P., Cardoso, J., Pereira Rodrigues, P., Riaño, D. (eds.) *Artificial Intelligence in Medicine*, pp. 338–348. Springer International Publishing, Cham (2021)



Convergence of a L_2 Regularized Policy Gradient Algorithm for the Multi Armed Bandit

Ștefana-Lucia Anița¹  and Gabriel Turinici² 

¹ “Octav Mayer” Institute of Mathematics of the Romanian Academy,
Bd. Carol I 8, Iași 700505, Romania

² CEREMADE, Université Paris Dauphine - PSL, CNRS, Paris, France
gabriel.turinici@dauphine.fr
<https://turinici.com>

Abstract. Although Multi Armed Bandit (MAB) on one hand and the policy gradient approach on the other hand are among the most used frameworks of Reinforcement Learning, the theoretical properties of the policy gradient algorithm used for MAB have not been given enough attention. We investigate in this work the convergence of such a procedure for the situation when a L_2 regularization term is present jointly with the ‘softmax’ parametrization. We prove convergence under appropriate technical hypotheses and test numerically the procedure including situations beyond the theoretical setting. The tests show that a time dependent regularized procedure can improve over the canonical approach especially when the initial guess is far from the solution.

Keywords: Reinforcement Learning · Multi Armed Bandit · Stochastic Gradient Descent Algorithm · Policy Gradient · Regularized Policy Gradients · Proximal Policy Optimization

1 Introduction

Supported by impressive practical applications including game play (e.g., Go [17], computer games [13]), autonomous car driving [5], ChatGPT [14], healthcare [10,22], recommender systems [1] etc., the Reinforcement Learning is a promising area of active research today. Standing out among Reinforcement Learning frameworks, the Multi Armed Bandit (MAB in the sequel) [7,18] has been extensively used both for theoretical investigations and for applications. We will focus here on a specific procedure, the softmax parameterized policy gradient as in [19, section 2.8 and chap. 13]. We investigate its convergence in presence of L_2 regularization¹ and numerically explore the performance of this regularized framework.

¹ In the machine learning literature the regularization considered here is denoted L_2 while in the mathematics literature the L^2 notation is more often used; we use L_2 throughout the text but both mean the same thing.

The plan of the paper is as follows: in the rest of this section we briefly review the literature while in section 2 we give the first notations and definitions. Then in section 3 we prove the convergence under some technical hypotheses, followed in section 4 by some numerical tests that confirm the theoretical results and also go beyond it to regimes not covered by the theory. We close with a discussion in section 5.

1.1 Brief literature review

The policy gradient algorithms have shown impressive results for applications in reinforcement learning but it has been long recognized that some corrections are necessary to improve convergence; several well known procedures implementing such corrections are the log-barrier penalized REINFORCE algorithm [23], trust-region policy optimization TRPO [16] and the proximal policy optimizations (PPO, the OpenAI’s default reinforcement learning algorithm); all use a form of regularization, i.e. all seek to limit and control the policy updates by various methods. In this general setting we will focus here on a different type of regularization and will most specifically talk about Multi Armed Bandits.

While the policy gradient algorithms show interesting numerical performance, the theoretical investigations of the convergence for the MAB have only recently witnessed important advances. In [8] it is proven that stochastic gradients procedures converge with high probability for the general situation of linear quadratic regulators while Agarwal et al. gave in [2] theoretical results under the general framework of Markov processes and specifically proved the convergence under different policy parameterizations; on the specific case of softmax parameterization that we analyze here, they examine three algorithms addressing this issue. The initial approach involves straightforward policy gradient descent on the objective without alterations. The second method incorporates entropic regularization to prevent the parameters from growing excessively, thereby ensuring sufficient exploration. Lastly, they investigate the natural policy gradient algorithm and demonstrate a global optimality outcome independent of the distribution mismatch coefficient or dimension-specific factors. Recall that in contrast we study here the softmax parameterization with $L2$ regularization.

In a very recent paper [4] published online just months ago (at the time of writing) J. Bhandari and D. Russo discuss the softmax parametrization but focus on (we cite) “an idealized policy gradient update with access to exact gradient evaluations”. As a distinction, we will focus here on the non-exact gradient (which is the one usually implemented) but at the price of stronger hypotheses. Yet in another state-of-the-art research [11] the authors make three contributions; first they establish that, when employing the true gradient (i.e., without the stochasticity), policy gradient with a softmax parametrization converges at a rate of $O(1/t)$. Then they examine entropy-regularized policy gradient and demonstrate its accelerated convergence rate. Finally, by integrating the aforementioned outcomes they describe the mechanism through which entropy regularization enhances policy optimization.

Finally, some other relevant works include [21] that study more specifically the situations when deep neural networks are used, while [24] investigate the infinite-horizon setting with discounted factors through a new variant that uses a random roll-out horizon for the Monte Carlo estimation.

On a more general theoretical view, as mentioned earlier, our focus is on softmax parameterized policy gradients with $L2$ regularization. We will employ arguments similar to that used for the convergence of general stochastic gradient descent (as developed from the initial proposal of Robbins and Monro [15]). A good book on this subject is [6] while recent works giving information on the convergence of the SGD for non-convex functions are [9, 12]; for short self-contained proofs see [3, 20]. Note that classical SGD convergence results as in [6, Thms 1.2.1 or 1.3.1] need several hypotheses, for instance the uniqueness of the critical point (here not true), some boundedness conditions (here without any regularization the optimal H will have infinite values), a convenient Lyapunov functional (the obvious one has degenerate directions in this case), some boundedness for the trajectories [12] and so on. Nevertheless, this will still constitute the basis of our work that puts together estimations and proofs from the literature that were not invoked in this setting before.

2 The softmax parameterized policy gradient Multi Armed Bandit with $L2$ regularization

We describe here the softmax parameterized Multi Armed Bandit policy gradient algorithm to which we add a $L2$ regularization term. For a description of the original Multi Armed Bandit (MAB) we refer to [19]. In the classical Multi Armed Bandit problem, we have k arms indexed by a where $a = 1, 2, \dots, k$. Each arm a has an associated reward distribution with mean $q_*(a)$. A case often considered is when the reward is normally distributed with mean $q_*(a)$ and variance $\sigma(a)^2 = 1$ (see later for our hypotheses on R which are more general). At each time step t , an agent selects an arm A_t and observes a reward $R_t \sim R(A_t)$ sampled from the distribution of the selected arm A_t . The goal is to maximize the cumulative reward over a fixed number of time steps or iterations.

In the policy gradient algorithm with softmax parametrization, the agent maintains a parameterized policy Π_H , where H is a parameter vector called ‘preference vector’. The preference vector H defines the probability $\Pi_H(A)$ to act on the arm A through the softmax mapping :

$$\Pi_H(A) = \frac{e^{H(A)}}{\sum_{a=1}^k e^{H(a)}}. \tag{1}$$

The MAB with regularization is formulated as finding the optimal preference vector $H \in \mathbb{R}^k$ solution to :

$$\text{maximize}_{H \in \mathbb{R}^k} \mathcal{L}_\gamma(H), \tag{2}$$

where the functional \mathcal{L}_γ is defined as :

$$\mathcal{L}_\gamma(H) := \mathbb{E}_{A \sim \Pi_H} \left[R(A) - \frac{\gamma}{2} \|H\|^2 \right]. \tag{3}$$

Here $A \sim \Pi_H$ means that A is sampled from the discrete law Π_H ; γ is a positive constant that is seen as a $L2$ regularization coefficient. For convenience, we will sometimes omit the γ in the notation and write only

$$\mathcal{L}(H) \tag{4}$$

instead of $\mathcal{L}_\gamma(H)$. Note that this description is **different** from the classical MAB [19, section 2.8] by the presence of the regularization term $\frac{\gamma}{2}\|H\|^2$. To solve (2) the policy gradient approach prescribes the use of a gradient ascent stochastic algorithm which can be written :

$$H_{t+1}(a) = H_t(a) + \rho_t [(R_t - \bar{R}_t)(\mathbb{1}_{a=A_t} - \Pi_{H_t}(a)) - \gamma H_t(a)], \quad a = 1, \dots, k, \quad (5)$$

where R_t is the reward at time t , \bar{R}_t is the mean reward up to time t and ρ_t a time step or ‘learning rate’ (see next section for the precise choice of time scheduling). Although the formula (5) seems somehow far from a stochastic gradient applied to \mathcal{L} we recall that this is indeed the case in the lemma 1 below.

3 Theoretical convergence results

We first recall why the term multiplying ρ_t in the right hand side of equation (5) is indeed an unbiased estimation of $\nabla_H \mathcal{L}(H)$.

To do this we need to be careful with the probabilistic framework; consider the filtration \mathcal{F}_t corresponding to all information available up to time t . To go to $t + 1$ two things happen: first the arm A_t is sampled with the discrete distribution Π_{H_t} ; then a reward is sampled from the distribution $R(A_t)$ of the arm A_t . As we will need very detailed information on this sampling, we need to make clear what part of the sampling is independent of \mathcal{F}_t and what part is measurable. Of course, $A_i, i < t$ and $H_i, i \leq t$ are \mathcal{F}_t mesurables; but, since A_t ’s distribution depend on H_t it cannot be independent of \mathcal{F}_t as random variable. Nevertheless, in MAB sampling :

$$\mathbb{E}[\mathbb{1}_{a=A_t} | \mathcal{F}_t] = \Pi_{H_t}(a). \quad (6)$$

To explain such a relation, imagine that the operations at time t start with sampling some uniform variable U_t in $[0, 1)$ independent of \mathcal{F}_t and then, depending on the value of U_t a comparison is made with components of Π_{H_t} to decide what value A_t will take; this can be written $\{A_t = a\} = \{U_t \in [\sum_{b=1}^{a-1} \Pi_{H_t}(b), \sum_{b=1}^a \Pi_{H_t}(b)]\}$ with convention that the first sum is 0 when $a = 1$. This gives equation (6). Now, once A_t is chosen, the choice of the reward follows the same path: there is a part that is independent of the specific value of A_t , for instance one can draw another V_t uniform in $[0, 1]$ and attribute the reward based on the quantile of the A_t distribution.

We denote

$$q_*(a) = \mathbb{E}[R(a)], \forall a \leq k \quad (7)$$

which, considering the definition of R_t , means that

$$\mathbb{E}[R_t \mathbb{1}_{a=A_t} | \mathcal{F}_t] = q_*(a), \forall a \leq k. \quad (8)$$

The following hypothesis will be considered true from now on :

$$\text{there exists a constant } C_m > 0 \text{ such that : } \mathbb{E}[R(a)^2] \leq C_m, \forall a \leq k. \quad (9)$$

We also introduce some notations for the terms appearing in the right hand side of (5) :

$$u_t(a) := (R_t - \bar{R}_t)(\mathbb{1}_{a=A_t} - \Pi_{H_t}(a)), \quad (10)$$

$$g_t(a) := (R_t - \bar{R}_t)(\mathbb{1}_{a=A_t} - \Pi_{H_t}(a)) - \gamma H_t(a). \quad (11)$$

We first give a preliminary result which explains why the algorithm (5) fits within the general framework of Robbins and Monro [15].

Lemma 1. *Under hypotheses (8) and (9) :*

$$\mathbb{E}[g_t | \mathcal{F}_t] = \nabla_H \mathcal{L}(H)|_{H=H_t}. \quad (12)$$

Moreover, for some constant C_{q_*} only depending on q_* and C_m :

$$\mathbb{E}[\|g_t\|^2] \leq C_{q_*} + 2\gamma^2 \|H_t\|^2. \quad (13)$$

Remark 1. The relation (12) says in essence that (5) is a Robbins-Monro type stochastic gradient in the sense that the stochastic estimate g_t of the gradient $\nabla_H \mathcal{L}(H)(a)|_{H=H_t}$ is unbiased. On the contrary, (13) is a technical point that will be required latter.

Proof. Equality (12) : Of course, the gradient of the $L2$ regularization term $\frac{\gamma}{2} \|H\|^2$ is γH which explains its presence in the left hand side, i.e., in g_t . On the other hand, the baseline \bar{R}_t satisfies :

$$\mathbb{E}[\bar{R}_t(\mathbb{1}_{a=A_t} - \Pi_{H_t}(a)) | \mathcal{F}_t] = \bar{R}_t \cdot \mathbb{P}[A_t = a] - \bar{R}_t \Pi_{H_t}(a) = 0. \quad (14)$$

Only the gradient of the reward R remains to be computed; we proceed as in [19, Section 2.8] by recalling that from (8) it follows that $\mathbb{E}[R_t | \mathcal{F}_t] = \mathbb{E}[\sum_a R(a) \mathbb{1}_{a=A_t} | \mathcal{F}_t] = \sum_a q_*(a) \Pi_{H_t}(a)$ which implies

$$\mathcal{L}(H) = \langle q_*, \Pi_H \rangle - \frac{\gamma}{2} \|H\|^2. \quad (15)$$

To conclude, it is enough to invoke the formula of the derivatives of the softmax function $H \mapsto \Pi_H$:

$$\frac{\partial \Pi_H(a)}{\partial H(b)} = \Pi_H(a) (\mathbb{1}_{a=b} - \Pi_H(b)). \quad (16)$$

Estimation (13) : since $g_t = u_t - \gamma H_t$, we only have to prove a bound for $\mathbb{E}[\|u_t\|^2 | \mathcal{F}_t]$. First note that $|\mathbb{1}_{a=A_t} - \Pi_{H_t}(a)| \leq 1$ so we are left with finding a bound for $\mathbb{E}[\|R_t - \bar{R}_t\|^2]$; but from (9) :

$$\mathbb{E}[\|R_t - \bar{R}_t\|^2] \leq 2\mathbb{E}[\|R_t\|^2] + 2\mathbb{E}[\|\bar{R}_t\|^2] \leq 2C_m + 2\mathbb{E}[\|\bar{R}_t\|^2]. \quad (17)$$

On the other hand $\bar{R}_t = \frac{R_0 + \dots + R_{t-1}}{t}$ with all terms having bounded second order moment (by (9)) which shows that $\mathbb{E}[\|\bar{R}_t\|^2] \leq C_m$ hence the conclusion. \square

3.1 Fixed time step

We prove now the first result involving the $L2$ regularized MAB including the case when the time step is constant (but small enough to ensure convergence) and γ large enough.

Proposition 1. *Denote*

$$\mu := \gamma - (\max_a q_*(a) - \min_a q_*(a)). \tag{18}$$

Under the hypotheses (8) and (9) assume

$$\mu > 0. \tag{19}$$

Then :

1. *the function \mathcal{L} defined in (3) has a unique maximum H_* ;*
2. *For any $t \geq 0$ denote*

$$d_t = \mathbb{E} [\|H_t - H_*\|^2]. \tag{20}$$

Then there exist constants $c_2, c_3, c_4 > 0$ depending only on q_ such that for $c_1 = c_4\gamma^2, c_0 = c_2 + c_3\gamma^2$:*

$$d_{t+1} \leq (1 - \rho_t\mu + \rho_t^2c_1)d_t + \rho_t^2c_0. \tag{21}$$

3. *For any $\epsilon > 0$ there exists a $\rho_\epsilon > 0$ such that if $\rho_t = \rho < \rho_\epsilon$ then*

$$\limsup_{t \rightarrow \infty} \mathbb{E} [\|H_{t+1} - H_*\|^2] \leq \epsilon. \tag{22}$$

4. *Take ρ_t a sequence such that:*

$$\rho_t \rightarrow 0 \text{ and } \sum_{t \geq 1} \rho_t = \infty. \tag{23}$$

Then $d_t \rightarrow 0$, or equivalently

$$\lim_{t \rightarrow \infty} H_t \stackrel{L^2}{=} H_*. \tag{24}$$

Proof. Item 1: We first establish some estimates concerning the Hessian $\nabla_H^2 \mathcal{L}$; Take c to be a constant. We can write :

$$\begin{aligned} \nabla_H^2 \mathcal{L}(H) &= \nabla_H^2 (\mathcal{L}(H) - c) = \nabla_H^2 (\langle q_* - c, \Pi_H \rangle - \frac{\gamma}{2} \|H\|^2) \\ &= \nabla_H^2 (\langle q_* - c, \Pi_H \rangle) - \gamma I_k. \end{aligned} \tag{25}$$

On the other hand, if we iterate the equation (16) once more we obtain for $A, a, b \leq k$:

$$\begin{aligned} \frac{\partial^2 \Pi_H(A)}{\partial H(b) \partial H(a)} &= \Pi_H(A) (\mathbb{1}_{a=A} - \Pi_H(a)) (\mathbb{1}_{b=A} - \Pi_H(b)) \\ &\quad - \Pi_H(A) \Pi_H(a) (\mathbb{1}_{b=a} - \Pi_H(b)) \leq 2\Pi_H(A). \end{aligned} \tag{26}$$

From this we obtain for any \bar{H} and variations δH :

$$\begin{aligned} &\nabla_H^2 \langle q_* - c, \Pi_H \rangle \Big|_{H=\bar{H}} (\delta H, \delta H) \\ &= \sum_{A=1}^k (q_*(A) - c) \sum_{a,b=1}^k \frac{\partial^2 \Pi_H(A)}{\partial H(b) \partial H(a)} \Big|_{H=\bar{H}} \delta H(a) \delta H(b) \\ &= \sum_{A=1}^k (q_*(A) - c) \Pi_{\bar{H}}(A) [\langle \delta H(A) - \delta H, \Pi_{\bar{H}} \rangle^2 - \langle \delta H^2, \Pi_{\bar{H}} \rangle + \langle \delta H, \Pi_{\bar{H}} \rangle^2] \\ &\leq \max_A |q_*(A) - c| \cdot |\langle \delta H(A) - \delta H, \Pi_{\bar{H}} \rangle^2 - \langle \delta H^2, \Pi_{\bar{H}} \rangle + \langle \delta H, \Pi_{\bar{H}} \rangle^2|. \end{aligned} \tag{27}$$

Take now $c = \frac{\max_a q_*(a) + \min_a q_*(a)}{2}$; then

$$\max_a |q_*(a) - c| = \frac{\max_a q_*(a) - \min_a q_*(a)}{2} =: \frac{c_*}{2}, \quad (28)$$

where the second part is a notation; since by Cauchy $\langle \delta H^2, \Pi_{\bar{H}} \rangle - \langle \delta H, \Pi_{\bar{H}} \rangle^2 \geq 0$ the term $\langle \delta H(A) - \delta H, \Pi_{\bar{H}} \rangle^2 - \langle \delta H^2, \Pi_{\bar{H}} \rangle + \langle \delta H, \Pi_{\bar{H}} \rangle^2$ is the difference of two positive numbers so its absolute value is smaller than the largest of them. We will prove that each is smaller than $2\|\delta H\|^2$. Obviously $\langle \delta H^2, \Pi_{\bar{H}} \rangle - \langle \delta H, \Pi_{\bar{H}} \rangle^2 \leq \langle \delta H^2, \Pi_{\bar{H}} \rangle \leq \max_a \delta H(a)^2 \leq \|\delta H\|^2$. For the first term we look for an optimum of $\langle \delta H(A) - \delta H, \Pi_{\bar{H}} \rangle^2$ under the constraint $\|\delta H\|^2 = 1$ and, after some straightforward computations we obtain 2 (see Lemma 2 for a proof). Thus finally :

$$\nabla_H^2 \langle q_* - c, \Pi_H \rangle|_{H=\bar{H}}(\delta H, \delta H) \leq c_* \|\delta H\|^2. \quad (29)$$

It follows from the previous considerations that

$$\nabla_H^2 \mathcal{L}(H)|_{H=\bar{H}}(\delta H, \delta H) \leq (c_* - \gamma) \|\delta H\|^2. \quad (30)$$

Take $H \in \mathbb{R}^k$. Using Taylor's formula for $s \mapsto sH$ we obtain some \bar{H} on the segment $[0, H]$ such that :

$$\begin{aligned} \mathcal{L}(H) &= \mathcal{L}(0) + \langle \nabla_H \mathcal{L}(0), H \rangle + \frac{1}{2} \nabla_H^2 \mathcal{L}(H)|_{H=\bar{H}}(H, H) \\ &\leq \mathcal{L}(0) + \langle \nabla_H \mathcal{L}(0), H \rangle + (c_*/2 - \gamma/2) \|H\|^2. \end{aligned} \quad (31)$$

When $\gamma > c_*$ we obtain that $-\mathcal{L}$ is coercive at infinity thus by continuity we obtain the existence of an optimum. The uniqueness follows from the strict concavity of \mathcal{L} (see inequality (30)).

Item 2: We have

$$\begin{aligned} \mathbb{E} [\|H_{t+1} - H_*\|^2] &= \mathbb{E} [\|H_t - H_* + \rho_t g_t\|^2] \\ &= \mathbb{E} [\|H_t - H_*\|^2] + \rho_t^2 \mathbb{E} [\|g_t\|^2] + 2\rho_t \mathbb{E} [\langle H_t - H_*, g_t \rangle]. \end{aligned} \quad (32)$$

From (12)

$$\mathbb{E} [\langle H_t - H_*, g_t \rangle] = \mathbb{E} [\langle H_t - H_*, \nabla_H \mathcal{L}(H_t) \rangle].$$

Recall that since H_* is an optimum $\mathcal{L}(H_*) \geq \mathcal{L}(H_t)$; using a Taylor expansion for $s \mapsto sH_* + (1-s)H_t$ around H_t and using the same estimations as above for the Hessian we obtain

$$\begin{aligned} \mathbb{E} [\langle H_t - H_*, \nabla_H \mathcal{L}(H_t) \rangle] &\leq \mathbb{E} \left[\mathcal{L}(H_t) - \mathcal{L}(H_*) - \frac{\mu}{2} \|H_t - H_*\|^2 \right] \\ &\leq -\frac{\mu}{2} \mathbb{E} [\|H_t - H_*\|^2]. \end{aligned} \quad (33)$$

Combining all these estimations and using (13) to bound the term $\mathbb{E} [\|g_t\|^2]$ we obtain the inequality (21). For the rest of the proof we follow the proof of Thm. 1 in [20].

Item 3: When $\rho_t = \rho$ estimation (21) is written

$$d_{t+1} - \frac{\rho c_0}{\mu - \rho c_1} \leq (1 - \rho\mu + \rho^2 c_1) \left(d_t - \frac{\rho c_0}{\mu - \rho c_1} \right).$$

If $\rho < \min(1/\mu, \mu/2c_1)$, taking the positive part allows to write :

$$\left(d_{t+1} - \frac{\rho c_0}{\mu - \rho c_1} \right)_+ \leq \left(1 - \frac{\rho\mu}{2} \right) \left(d_t - \frac{\rho c_0}{\mu - \rho c_1} \right)_+,$$

and therefore $\forall \ell \geq 1$:

$$\left(d_{n+\ell} - \frac{\rho c_0}{\mu - \rho c_1} \right)_+ \leq \left(1 - \frac{\rho\mu}{2} \right)^\ell \left(d_t - \frac{\rho c_0}{\mu - \rho c_1} \right)_+.$$

For $\ell \rightarrow \infty$ we obtain $\limsup_\ell \left(d_\ell - \frac{\rho c_0}{\mu - \rho c_1} \right)_+ = 0$ which gives the conclusion (22) for $\rho \leq \rho_\epsilon := \min\{1/\mu, \mu/2c_1, \epsilon\mu/(c_0 + \epsilon c_1)\}$.

Item 4: Consider now ρ_t non-constant and fix $\epsilon > 0$; we invoke inequality (21) and obtain :

$$d_{t+1} - \epsilon \leq \left(1 - \frac{\rho_t \mu}{2} \right) (d_t - \epsilon) + \rho_t (c_0 \rho_t - \mu \epsilon / 2 + (\rho_t c_1 - \mu / 2) d_t).$$

When t is big enough, the last term in the right hand side is negative and therefore

$$d_{t+1} - \epsilon \leq \left(1 - \frac{\rho_t \mu}{2} \right) (d_t - \epsilon),$$

hence

$$(d_{t+1} - \epsilon)_+ \leq \left(1 - \frac{\rho_t \mu}{2} \right) (d_t - \epsilon)_+.$$

Taking the product of all relations of this type allows to write :

$$(d_{t+\ell} - \epsilon)_+ \leq \prod_{s=t}^{t+\ell-1} \left(1 - \frac{\rho_s \mu}{2} \right) (d_t - \epsilon)_+. \tag{34}$$

Using the Lemma 2 from [20] recalled as Lemma 4 below we obtain $\lim_{\ell \rightarrow \infty} (d_\ell - \epsilon)_+ = 0$ and since this is true for any ϵ the conclusion follows. \square

Lemma 2. Let $\Pi \in \mathbb{R}^k$, $\Pi(a) \geq 0, \forall a \leq k, \sum_a \Pi(a) = 1$. Then for any $x \in \mathbb{R}^k$ and any $\ell \leq k$

$$\langle x, \Pi \rangle - x_\ell \leq 2\|x\|^2. \tag{35}$$

Proof. The term $\langle x, \Pi \rangle$ is a mean value of x under the law Π thus it is somewhere between the smallest (denoted x_m) and the largest (denoted x_M) values of $x_i, i \leq k$. The left hand side is thus smaller than $(x_M - x_m)^2$. On the other hand, $2\|x\|^2 \geq 2(x_m^2 + x_M^2) \geq (x_M - x_m)^2$. \square

3.2 Convergence rates for linear decay $\rho_t = \frac{\beta_1}{1+\beta_2 t}$ and large γ

We investigate now the situation when ρ_t is not constant but decays linearly.

Proposition 2. *Let $\beta_1, \beta_2 > 0$ two positive constants and take*

$$\rho_t = \frac{\beta_1}{1 + \beta_2 t}. \quad (36)$$

Under the hypotheses (8) and (9) for γ large enough :

1. *the problem (2) has a unique solution H_* ;*
2. *the L2 regularized policy gradient MAB algorithm (5) converges with the rate :*

$$\mathbb{E}[\|H_t - H_*\|^2] = O\left(\frac{1}{t}\right) \text{ as } t \rightarrow \infty. \quad (37)$$

Proof. We saw already in proposition 1 that the optimum exists and is unique for $\mu > 0$. We also saw that (21) is satisfied. Denote $\xi_t = td_t$. By multiplication with $t + 1$ inequality (21) can be written in terms of ξ as

$$\xi_{t+1} \leq (1 - \rho_t \mu + c_1 \rho_t^2) \xi_t \left(1 + \frac{1}{t}\right) + c_0 \rho_t^2 (t + 1). \quad (38)$$

It is enough to prove that ξ_t is bounded to conclude. Suppose on the contrary that ξ_t is not bounded. In this case, for any C large enough there exists some rank t_C large enough where ξ_{t+1} is for the first time larger than C . In particular this means that $\xi_t \leq C \leq \xi_{t+1}$. Therefore

$$C \leq \xi_{t+1} \leq (1 - \rho_t \mu + c_1 \rho_t^2) C \left(1 + \frac{1}{t}\right) + c_0 \rho_t^2 (t + 1). \quad (39)$$

so finally

$$C \leq (1 - \rho_t \mu + c_1 \rho_t^2) C \left(1 + \frac{1}{t}\right) + c_0 \rho_t^2 (t + 1), \quad (40)$$

or, after simplification by C in both terms and multiplication by t :

$$0 \leq C [t(-\rho_t \mu + c_1 \rho_t^2) + 1 - \rho_t \mu + c_1 \rho_t^2] + c_0 \rho_t^2 (t + 1)t. \quad (41)$$

Recall that $\rho_t = \frac{\beta_1}{1+\beta_2 t}$. For $t \rightarrow \infty$ the term $c_0 \rho_t^2 (t + 1)t$ tends to the constant $c_0 \frac{\beta_1^2}{\beta_2^2}$. The terms multiplying C tends to

$$\lim_{t \rightarrow \infty} [t(-\rho_t \mu + c_1 \rho_t^2) + 1 - \rho_t \mu + c_1 \rho_t^2] = 1 - \mu \frac{\beta_1}{\beta_2}. \quad (42)$$

But for μ large enough (i.e., γ large enough) this is negative so the right hand side in (41) cannot remain positive when t and C are large enough because is a sum of a bounded term and a product between C and a quantity that converges to a strictly negative constant. This provides the required contradiction and ends the proof. \square

3.3 Behavior when $\gamma \rightarrow 0$

For completeness, we investigate in this section the other side of the question, namely the regularization part.

The real goal is to solve problem (2) for $\gamma = 0$. When $\gamma > 0$ the solution of the problem (2) will not coincide with the solution for $\gamma = 0$. The question is whether this perturbation will be small when γ is small. This is an intuitive result but not completely trivial because when $\gamma = 0$ the maximum in (2) is not attained in general as most of its components will tend to $-\infty$. Indeed, suppose all $q_*(a)$ are different and denote by $q_*(a_{max})$ the largest one. When $\gamma = 0$ the functional is simply $\mathcal{L}_0(H) = \sum_a q_*(a) \Pi_H(a) < q_*(a_{max})$. The inequality is always strict but $q_*(a_{max}) - \mathcal{L}_0(H)$ vanishes when Π_H is a Dirac mass in a_{max} ; for that to happen H would have to have all entries equal to $-\infty$ except $H_{a_{max}}$ that can be any finite value.

The result below informs that, as expected, we can be as close as we want to the optimum value of the non-regularized MAB problem by taking γ small enough.

Lemma 3. *Let*

$$V(\gamma) := \max_{H \in \mathbb{R}^k} \mathcal{L}_\gamma(H). \tag{43}$$

Then $\lim_{\gamma \rightarrow 0} V(\gamma) = V(0)$.

Proof. For any $\gamma > 0$ denote H_γ^* one optimum in (2). Note first that, by standard coercivity and continuity on compacts arguments of $-\mathcal{L}_\gamma$, this optimum value exists for any $\gamma > 0$ (it is not necessarily unique though); for $\gamma = 0$ it does not exist in general as the maximum value is attained only as a limit of values along a sequence H_n . Take H_n to be a sequence such that

$$\lim_{n \rightarrow \infty} \mathcal{L}_0(H_n) = \sup_{H \in \mathbb{R}^k} \mathcal{L}_0(H) = V(0). \tag{44}$$

By the very definition of H_γ^* :

$$V(\gamma) = \mathcal{L}_\gamma(H_\gamma^*) \geq \mathcal{L}_\gamma(H), \quad \forall H \in \mathbb{R}^k. \tag{45}$$

In particular

$$V(\gamma) = \mathcal{L}_\gamma(H_\gamma^*) \geq \mathcal{L}_\gamma(H_n) = \mathcal{L}_0(H_n) - \frac{\gamma}{2} \|H_n\|^2, \quad \forall n \geq 1. \tag{46}$$

Keep now n fixed and let $\gamma \rightarrow 0$ we obtain $\liminf_{\gamma \rightarrow 0} V(\gamma) \geq \mathcal{L}_0(H_n)$. Take now $n \rightarrow \infty$ to obtain $V(0) \leq \liminf_{\gamma \rightarrow 0} V(\gamma)$. But since on the other hand for all $\gamma \geq 0$: $V(\gamma) \leq V(0)$ we obtain the conclusion. \square

4 Numerical simulations

The Python implementation is available on Github ². We perform $M = 1000$ tests of 2000 steps each; for each of the M tests we sample, as in [19, fig 2.5 page 38] $k = 10$

² https://github.com/gabriel-turinici/regularized_policy_gradient version August 31st 2024.

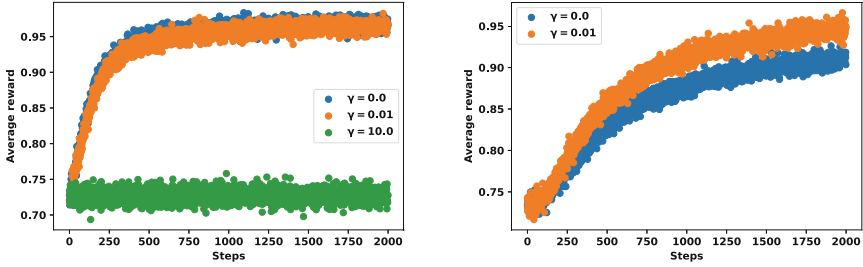


Fig. 1. The average reward for $\rho_t = 0.05$ (constant), γ is 0, 0.01 or 10 (see the legend). **Left :** start from a uniform distribution Π_{H_0} with $H_0 = (0, \dots, 0)$. **Right :** start from a biased distribution Π_{H_0} with $H_0 = (5, \dots, 0)$.

arms with $q_*(a)$, $a = 1, \dots, k$ independent and normally distributed with mean 4 and unit variance.

Once $q_*(\cdot)$ have been sampled they do not change for the 2000 steps of the respective test. To ensure fair comparison we use same values of $q_*(\cdot)$ for all the bandits that are compared, for instance in figure 1 run number 123 for $\gamma = 0$ and run number 123 for $\gamma = 0.01$ and run number 123 for $\gamma = 10$ share the same $q_*(\cdot)$, which is different from the $q_*(\cdot)$ of runs 122. For each of the arms $a = 1, \dots, k$ the law of $R(A)$ conditional to $A = a$ is a normal variable with mean $q_*(a)$ and unit variance. Note that in this case

$$c_*^{avg} := \mathbb{E}[\max_{a \leq k} q_*(a) - \min_{a \leq k} q_*(a)] \simeq 3.08. \quad (47)$$

The proposition 1 prescribes that γ should be larger than c_*^{avg} .

The uniform distribution corresponding to $H_0 = (0, \dots, 0)$ would give, in average, a reward equal to 4. Nevertheless in the following, for each of the M tests we will not plot the absolute value of the reward but the value relative to the maximum possible reward $\max_{a \leq k} q_*(a)$ (because this maximum varies with each run). With this convention the best possible reward is 1. The average over the $M = 1000$ runs are presented in figure 1 and discussed also in section 5. We see that when starting for the uniform distribution the regularization $\gamma = 0.01$ does not prevent the algorithm to have a performance comparable with the non-regularized version (i.e., $\gamma = 0$). On the contrary, the value $\gamma = 10$ is too large and biases the algorithm towards a non-optimal solution (similar results are obtained for $\gamma = 3.08$). When starting from a biased distribution, the regularization $\gamma = 0.01$ does a better job and obtains a visible improvement over the performance of the non-regularized version (i.e., $\gamma = 0$).

Additional tests are presented in figures 2 and 3 where we investigate a linear decay schedule for the learning rate ρ_t and also for the regularization coefficient γ_t . In particular in figure 3 it is seen that the non-null initial regularization helps leaving the non-optimal initial guess H_0 ; then the decay of γ_t will provide results comparable the non-regularized version in particular convergence to an optimal point.

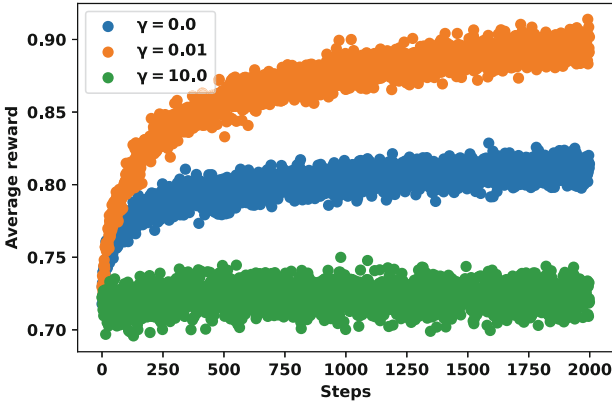


Fig. 2. The average reward when starting from the non-uniform distribution Π_{H_0} with $H_0 = (5, \dots, 0)$ and $\rho_t = \frac{1}{1+0.05*t}$ in the general setting of proposition 2 equation (36); we test $\gamma = 0$, $\gamma = 0.01$ or $\gamma = 10$ (see the legend). As before, $\gamma = 10$ is too large to obtain good results.

5 Summary and discussion

We considered in this work a L_2 regularized policy gradient algorithm applied to a Multi Armed Bandit (MAB) and investigated it both theoretically (convergence, rate of convergence) and numerically. Let us first recall that it was already remarked in the literature [11] that the MAB may behave erratically when the initialization is close to a sub-optimal critical point (there are many of them, for instance all Dirac masses are critical points). In this case the standard gradient policy MAB will spend a long time in this region before converging to the global maximum. One way to cure this drawback is to introduce regularization, in our case this is L_2 regularization, parameterized by a multiplicative coefficient γ .

Under technical conditions on the value of γ we gave two convergence results : proposition 1 that works for both constant and variable time steps ρ_t and proposition 2 that proves that the convergence happens at rate $O(1/t)$ if ρ_t decays linearly. However the existence of the regularization part (when $\gamma > 0$) may shift the optimal solution; we proved then in lemma 3 that when $\gamma \rightarrow 0$ the optimality is restored.

The technical conditions in the theoretical results impose large values of γ but in practice small values of γ are requested for good quality solution. To see the usefulness of the regime when γ is small, we tested the procedure numerically. The results indicate that irrespective of whether γ is large or small the convergence occurs when the initial guess H_0 is uniform, but the quality of the optimum is not good when γ is too large, see figure 1; same holds true when non-constant (linear decay) ρ_t is used, see figure 2; when the initial guess H_0 is **not uniform** the convergence is significantly better with $\gamma > 0$ and for γ not too large its quality is also very good, see figure 1.

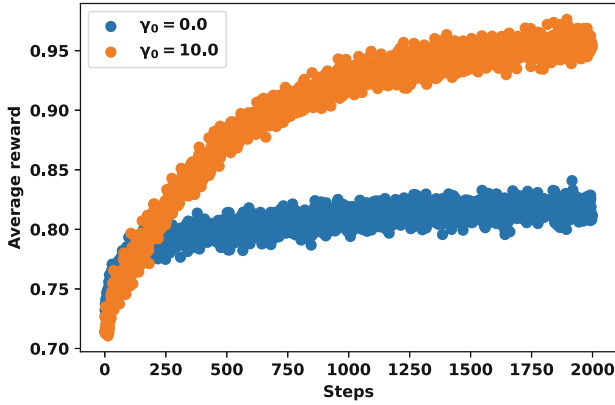


Fig. 3. The average reward when starting from the biased distribution Π_{H_0} with $H_0 = (5, \dots, 0)$ and $\gamma_t = \frac{\gamma_0}{1+0.2*t}$ (see the legend), $\gamma_0 = 0$ (no regularization) or $\gamma_0 = 10$. We take $\rho_t = \frac{1}{1+0.05*t}$ (see eq. (36)).

To combine the best of the two possible worlds, we also tested a variable γ_t of the form $\gamma_t = \frac{\gamma_0}{1+\eta t}$ (with η a positive constant) starting from non-uniform initial guess H_0 ; the results in figure 3, are very good and show that this choice is better than the classical non regularized policy gradient procedure. The precise optimal decay schedule for γ_t is not known and will be left for future works.

A Appendix

For completeness we recall below the Lemma 2 from [20] and its proof.

Lemma 4. *Let $\xi > 0$ and ρ_t a sequence of positive real numbers such that $\rho_t \rightarrow 0$ and $\sum_{t \geq 1} \rho_t = \infty$. Then for any $t \geq 0$:*

$$\lim_{\ell \rightarrow \infty} \prod_{j=t}^{t+\ell} (1 - \rho_j \xi) = 0. \tag{48}$$

Proof. Since $\rho_t \rightarrow 0$, $\rho_j \xi < 1$ for j large enough; without loss of generality we can suppose this is true starting from t . Since for any $x \in]0, 1[$ we have $\log(1 - x) \leq -x$:

$$0 \leq \prod_{j=t}^{t+\ell} (1 - \rho_j \xi) = e^{\sum_{j=t}^{t+\ell} \log(1 - \rho_j \xi)} \leq e^{\sum_{j=t}^{t+\ell} (-\rho_j \xi)} \xrightarrow{\ell \rightarrow \infty} e^{-\infty} = 0. \tag{49}$$

□

B Further comments on the assumption $\mu > 0$

The convergence of the scheme proved in proposition 1 requires $\mu := \gamma - c_* > 0$ (c_* is defined in (28)). If for some reason a γ is given and cannot be changed and $\gamma - c_* \leq 0$ we can still get convergence if we consider the modified softmax parametrized MAB policy gradient algorithm that finds $H \in \mathbb{R}^k$ solution to (2) where the functional \mathcal{L}_γ is replaced by : $\mathbb{E}_{A \sim \Pi_H^\alpha} [R(A) - \frac{\gamma}{2} \|H\|^2]$ where $\Pi_H^\alpha(A) := \frac{e^{\alpha H(A)}}{\sum_{a=1}^k e^{\alpha H(a)}}$ i.e., $\Pi_H^\alpha = \Pi_{\alpha H}$. Here $\alpha > 0$ is an arbitrary but fixed constant such that $\gamma - \alpha^2 c_* > 0$. Note that the only change is the replacement of Π_H by Π_H^α . With this provision the stochastic gradient ascent algorithm (5) is replaced by :

$$H_{t+1}(a) = H_t(a) + \rho_t \left[\alpha (R_t - \bar{R}_t) (\mathbb{1}_{a=A_t} - \Pi_{H_t}^\alpha(a)) - \gamma H_t(a) \right], \quad a = 1, \dots, k. \quad (50)$$

The proof of the Proposition 1 remains the same as soon as we replace μ in (18) with $\gamma - \alpha^2 c_* > 0$ and work with

$$u_t := \alpha (R_t - \bar{R}_t) (\mathbb{1}_{a=A_t} - \Pi_{H_t}^\alpha(a)) \quad (51)$$

$$g_t := \alpha (R_t - \bar{R}_t) (\mathbb{1}_{a=A_t} - \Pi_{H_t}^\alpha(a)) - \gamma H_t(a). \quad (52)$$

The proofs of Lemma 1 and Proposition 1 will use

$$\frac{\partial \Pi_H^\alpha(a)}{\partial H(b)} = \alpha \Pi_H^\alpha(a) (\mathbb{1}_{a=b} - \Pi_H^\alpha(b)) \quad (53)$$

and

$$\begin{aligned} \frac{\partial^2 \Pi_H^\alpha(A)}{\partial H(b) \partial H(a)} &= \alpha^2 \Pi_H^\alpha(A) (\mathbb{1}_{a=A} - \Pi_H^\alpha(a)) (\mathbb{1}_{b=A} - \Pi_H^\alpha(b)) \\ &\quad - \alpha^2 \Pi_H^\alpha(A) \Pi_H^\alpha(a) (\mathbb{1}_{b=a} - \Pi_H^\alpha(b)) \leq 2\alpha^2 \Pi_H^\alpha(A). \end{aligned} \quad (54)$$

Note however that for given α, γ this modified procedure will converge to the same optimal H_* as the original procedure (i.e. $\alpha = 1$) if we take γ/α^2 instead of the original γ (direct computations show that they share the same critical point equations).

References

1. Afsar, M.M., Crump, T., Far, B.: Reinforcement learning based recommender systems: A survey. *ACM Comput. Surv.* **55**(7) (December 2022). <https://doi.org/10.1145/3543846>, <https://doi.org/10.1145/3543846>
2. Agarwal, A., Kakade, S.M., Lee, J.D., Mahajan, G.: Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes. In: Abernethy, J., Agarwal, S. (eds.) *Proceedings of Thirty Third Conference on Learning Theory. Proceedings of Machine Learning Research*, vol. 125, pp. 64–66. PMLR (09–12 Jul 2020), <https://proceedings.mlr.press/v125/agarwal20a.html>
3. Anița, Ş.-L., Turinici, G.: On the Convergence Rate of the Stochastic Gradient Descent (SGD) and Application to a Modified Policy Gradient for the Multi Armed Bandit (2024), [arxiv:2402.06388](https://arxiv.org/abs/2402.06388)

4. Bhandari, J., Russo, D.: Global Optimality Guarantees for Policy Gradient Methods. *Operations Research* (Jan 2024). <https://doi.org/10.1287/opre.2021.0014>, <https://pubsonline.informs.org/doi/full/10.1287/opre.2021.0014>, publisher: INFORMS
5. Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K.: End to end learning for self-driving cars (2016), [arxiv:1604.07316](https://arxiv.org/abs/1604.07316)
6. Chen, H.F.: Stochastic approximation and its applications, *Nonconvex Optim. Appl.*, vol. 64. Dordrecht: Kluwer Academic Publishers (2002)
7. Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Gordon, G., Dunson, D., Dudík, M. (eds.) *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 15, pp. 208–214. PMLR, Fort Lauderdale, FL, USA (11–13 April 2011), <https://proceedings.mlr.press/v15/chu11a.html>
8. Fazel, M., Ge, R., Kakade, S., Mesbahi, M.: Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 1467–1476. PMLR (Jul 2018), <https://proceedings.mlr.press/v80/fazel18a.html>
9. Fehrman, B., Gess, B., Jentzen, A.: Convergence rates for the stochastic gradient descent method for non-convex objective functions. *Journal of Machine Learning Research* **21**(136), 1–48 (2020), <http://jmlr.org/papers/v21/19-636.html>
10. Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., Celi, L.A.: Guidelines for reinforcement learning in healthcare. *Nat. Med.* **25**(1), 16–18 (2019)
11. Mei, J., Xiao, C., Szepesvari, C., Schuurmans, D.: On the Global Convergence Rates of Softmax Policy Gradient Methods. In: III, H.D., Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 119, pp. 6820–6829. PMLR (Jul 2020), <https://proceedings.mlr.press/v119/mei20b.html>
12. Mertikopoulos, P., Hallak, N., Kavis, A., Cevher, V.: On the Almost Sure Convergence of Stochastic Gradient Descent in Non-Convex Problems. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1117–1128. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/0cb5ebb1b34ec343dfe135db691e4a85-Paper.pdf, [arxiv:2006.11144](https://arxiv.org/abs/2006.11144)
13. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (Feb 2015). <https://doi.org/10.1038/nature14236>
14. OpenAI: Gpt-4 technical report (2024), [arxiv:2303.08774](https://arxiv.org/abs/2303.08774)
15. Robbins, H., Monro, S.: A Stochastic Approximation Method. *The Annals of Mathematical Statistics* **22**(3), 400 – 407 (1951). <https://doi.org/10.1214/aoms/117729586>, publisher: Institute of Mathematical Statistics
16. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: Bach, F., Blei, D. (eds.) *Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 37, pp. 1889–1897. PMLR, Lille, France (07–09 Jul 2015), <https://proceedings.mlr.press/v37/schulman15.html>
17. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. *Nature* **529**(7587), 484–489 (Jan 2016). <https://doi.org/10.1038/nature16961>

18. Slivkins, A.: Introduction to Multi-Armed Bandits. Foundations and Trends® in Machine Learning **12**(1-2), 1–286 (2019). <https://doi.org/10.1561/22000000068>
19. Sutton, R.S., Barto, A.G.: Reinforcement learning. An introduction. Adapt. Comput. Mach. Learn., Cambridge, MA: MIT Press, 2nd expanded and updated edition edn. (2018)
20. Turinici, G.: The convergence of the Stochastic Gradient Descent (SGD) : a self-contained proof (2023). <https://doi.org/10.5281/ZENODO.4638694>, [arxiv:2103.14350v2](https://arxiv.org/abs/2103.14350v2)
21. Wang, L., Cai, Q., Yang, Z., Wang, Z.: Neural policy gradient methods: Global optimality and rates of convergence (2019), [arxiv:1909.01150](https://arxiv.org/abs/1909.01150)
22. Yu, C., Liu, J., Nemati, S., Yin, G.: Reinforcement learning in healthcare: A survey. ACM Computing Surveys (CSUR) **55**(1), 1–36 (2021)
23. Zhang, J., Kim, J., O’Donoghue, B., Boyd, S.: Sample efficient reinforcement learning with reinforce. Proceedings of the AAAI Conference on Artificial Intelligence **35**(12), 10887–10895 (May 2021). <https://doi.org/10.1609/aaai.v35i12.17300>, <https://ojs.aaai.org/index.php/AAAI/article/view/17300>
24. Zhang, K., Koppel, A., Zhu, H., Başar, T.: Global Convergence of Policy Gradient Methods to (Almost) Locally Optimal Policies. SIAM Journal on Control and Optimization **58**(6), 3586–3612 (2020). <https://doi.org/10.1137/19M1288012>, eprint: <https://doi.org/10.1137/19M1288012>



Delving into Feature Space: Improving Adversarial Robustness by Feature Spectral Regularization

Zhen Cheng^{1,2}, Fei Zhu³, Xu-Yao Zhang^{1,2}, and Cheng-Lin Liu^{1,2}(✉)

¹ State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA, Beijing, China

chengzhen2019@ia.ac.cn, xyz@nlpr.ia.ac.cn

² School of Artificial Intelligence, UCAS, Beijing, China

liucl@nlpr.ia.ac.cn

³ Centre for Artificial Intelligence and Robotics, HKISI-CAS, Hong Kong, China

zhufei2018@ia.ac.cn

Abstract. The study of adversarial examples in deep neural networks has attracted great attention. Numerous methods improve adversarial robustness via shrinking the gap of features between natural examples and adversarial examples. Nevertheless, the role of individual features in adversarial robustness has not been explored adequately. In this paper, we delve into this problem from the perspective of spectral analysis in feature space. We find that while standardly trained deep models have features distributed dominantly along eigenvectors with large eigenvalues, eigenvectors with smaller eigenvalues are more sensitive to adversarial attacks. We attribute this phenomenon to the dominance of the top eigenvalues, linked to the concept of intrinsic dimensionality. The extracted features possess a small intrinsic dimensionality, enhancing generalization but resulting in the model overlooking diverse features. We propose a method called *Feature Spectral Regularization (FSR)* to penalize the largest eigenvalue, so as to spread the distribution of eigenvalues. Comprehensive experiments demonstrate that FSR is effective to alleviate the dominance of larger eigenvalues, increase the intrinsic dimensionality, and improve adversarial robustness on multiple datasets.

Keywords: Adversarial example · adversarial robustness · spectral analysis · distribution of eigenvalues · feature spectral regularization

1 Introduction

It is shown that the performance of Deep Neural Networks (DNNs) decreases dramatically when confronted with adversarial examples [5, 13, 29]. To mitigate

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_28.

this vulnerability of DNNs, numerous methods have been proposed to improve adversarial robustness [21,34]. Among them, adversarial training (AT) [21] has achieved the state-of-the-art performance under various attacks [10]. Different from standard training, adversarial training introduces adversarial examples into DNN training to improve the adversarial robustness.

One primary aim of adversarial training is to eliminate the gap of features between natural examples and adversarial examples [40]. However, without considering the distinction of contribution of individual features, this gap cannot be addressed adequately. This is important since every feature may play a different role in robustness. The work of [17] argued that adversarial examples result from non-robust features [17], *i.e.*, those generalize well in standard classification but are brittle to adversary. This inspires us to measure the robustness of individual features. While the work of [3,37] considered the influence of different channels of DNNs on robustness, the spectral characteristics of features, *i.e.*, the eigenvalues and eigenvectors of feature covariance, have not been explored clearly.

In this paper, we analyze the connection between the spectral components of deep features and adversarial robustness. Through applying principal component analysis (PCA) to deep features, we split feature space into components corresponding to different eigenvectors and eigenvalues. It is observed that standardly trained models often results in a sharp distribution of eigenvalues [38], *i.e.*, the eigenvalues rapidly decrease along the component dimension, as shown in Fig. 1. This property may be beneficial for dimensionality reduction for improving the classification accuracy in standard setting [20,24], but it is yet unclear how to exploit the spectral property for adversarial robustness. We hypothesize that the sharp distribution of eigenvalues implies that deep models have learned less diverse features while ignoring the vulnerability of features to adversary. While a minority of eigenvalues occupying the overwhelming majority in the sum of eigenvalues generalize well for standard classification, the eigenvectors with smaller eigenvalues affecting adversarial robustness, are likely to be omitted. To verify our hypothesis, we define a new metric to measure the variation of features along different eigenvectors under attacks, as shown in Fig. 3~4. Our observation reveals that the adversary tends to project on more components along the eigenvectors with smaller eigenvalues, and the variation of eigenvalues can be alleviated by AT. This phenomenon could also be connected with intrinsic dimensionality. The extracted features possess a small intrinsic dimensionality, enhancing generalization but resulting in the model overlooking diverse features.

Therefore, we propose to improve adversarial robustness by alleviating the sharp distribution of eigenvalues. To guide the spectrum of eigenvalues distributes on more components, we propose a regularizer named Feature Spectral Regularization (FSR) to penalize the largest eigenvalue of the feature matrix covariance. Empirical studies show that FSR spreads the overall distribution of eigenvalues, making models focus on more spectral components. We also provide a theoretical explanation based on robust linear regression. Comprehensive

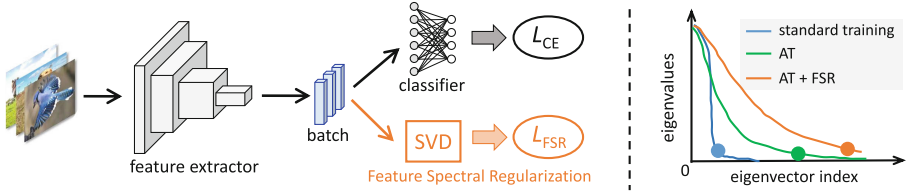


Fig. 1. The architecture of DNN with Feature Spectral Regularization (FSR). FSR alleviates the dominance of top eigenvalues and enhances the role of relatively smaller eigenvalues. When combined with AT, FSR helps learn more diverse features, increasing the intrinsic dimensionality (denoted by the solid circles).

experiments confirm that FSR indeed improves the adversarial robustness on multiple datasets. Our contributions are summarized as follows:

- We find a close connection between the spectral property of features and adversarial robustness. On one hand, standardly trained models produce a sharp distribution of eigenvalues, which is beneficial for classification accuracy in standard setting while harmful in adversarial setting. On the other hand, the adversary tends to project more on eigenvectors with smaller eigenvalues.
- We propose Feature Spectral Regularization (FSR) to spread the overall distribution of eigenvalues in deep feature space, so as to improve the adversarial robustness of DNNs, and a theoretical explanation based on robust linear regression is provided.
- We verify that FSR is effective in improving adversarial robustness and alleviating the sharp distribution of eigenvalues, by comprehensive experiments.

2 Related Work

Adversarial Defense. Many defense methods have been proposed to improve adversarial robustness since the discovery of adversarial examples [8, 34]. However, many of them are proven to be non-effective because they highly depend on obfuscated gradients [2]. Among these, adversarial training [21] is now regarded as the state-of-the-art method [22, 27]. Distinguished from standard training, adversarial training trains DNN on adversarial examples:

$$\min_{\theta} \mathbb{E}_{(x,y) \in \mathcal{D}} \max_{\tilde{x} \in \mathbb{B}(x, \epsilon)} L_{CE}(\tilde{x}, y; \theta), \tag{1}$$

where \mathcal{D} is the training dataset, the parameters of DNN are denoted as θ , L_{CE} is the cross-entropy (CE) loss, and $\mathbb{B}(x, \epsilon) = \{\tilde{x} : \|\tilde{x} - x\| \leq \epsilon\}$ means the constraint of perturbation in ϵ -ball. Furthermore, Projected Gradient Descent (PGD) [21] is often used to generating adversarial examples in AT:

$$\tilde{x} \leftarrow \Pi_{\mathbb{B}(x, \epsilon)}(\tilde{x} + \eta \cdot \text{sign}(\nabla_{\tilde{x}} L_{CE}(\tilde{x}, y; \theta))), \tag{2}$$

where Π is the projection function and η is the step size of PGD.

Based on adversarial training, some variants with new objective functions or regularizations have been proposed. By introducing a trade-off between robustness and generalization, TRADES [40] reaches comparative robustness with AT:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}} \{L_{\text{CE}}(\mathbf{x}, y; \theta) + \beta \max_{\tilde{\mathbf{x}} \in \mathbb{B}(\mathbf{x}, \epsilon)} D_{\text{KL}}(f_{\theta}(\mathbf{x}) \| f_{\theta}(\tilde{\mathbf{x}}))\}, \quad (3)$$

where $D_{\text{KL}}(\cdot \| \cdot)$ is the Kullback-Leibler divergence, β is the robustness regularization, and f_{θ} is the score function that maps an instance to the output distribution (softmax of logits).

There are some works that build upon TRADES and AT. Gowal *et al.* [14] found that training with generated data can enhance robustness. Adversarial Weight Perturbation (AWP) [33] explicitly regularizes the flatness of weight loss landscape, and forms a double-perturbation mechanism. Some works also attempt to analyze the impact of AT on both individual samples [18] and categories [32]. However, apart from these, it is still not well understood how adversarial training boosts adversarial robustness from the perspective of spectral properties.

Spectral Properties of Feature Representations. Some studies have revealed that the spectral properties of features influence the performance in various learning tasks. For example, the spectral properties are crucial to detect backdoors [15]. The eigenvectors corresponding to the larger eigenvalues are found to dominate the transferability of features in domain adaptation [9]. By utilizing the principle of Maximal Coding Rate Reduction, it is theoretically proven that the larger several singular values of feature matrix for every class should be equal to learn the maximally diverse representation [38]. Some works also analyze neural networks' spectral properties from a theoretical perspective [26, 28, 31], such as analyzing the properties of the Jacobian matrix.

Different from these studies, we analyse the connection between adversarial robustness and spectral components of deep features. We aim to explore which components are more fragile under attacks, and propose a method to boost adversarial robustness by constraining spectral properties.

3 Spectral Analysis in Feature Space

In this section, we investigate the connection between spectral properties and adversarial robustness.

3.1 Curve of Eigenvalues and Adversarial Robustness

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ including C classes, \mathbf{x}_i represents the input data and y_i is the label. DNN is composed of a feature extractor $h(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}^d$ and a linear classifier $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^C$. After centralizing the learned features (*i.e.* $\frac{1}{n} \sum_{i=1}^n h(\mathbf{x}_i) = 0$), we decompose the learned features by spectral decomposition:

$$\frac{1}{n} \sum_{i=1}^n h(\mathbf{x}_i) h(\mathbf{x}_i)^{\text{T}} = \sum_{j=1}^d \mathbf{u}_j \lambda_j \mathbf{u}_j^{\text{T}}, \quad (4)$$

where λ_j means the eigenvalues with index j and $\mathbf{u}_j \in \mathbb{R}^d$ represents its eigenvector. We choose a popularly used architecture, ResNet-18 [16], to be trained using both standard training and adversarial training on CIFAR-10 [19]. The parameters for AT are the same as that in [27]. We calculate the eigenvalues by applying Eq. (4), and plot them in Fig. 2. The features come from the penultimate layer (512 dimensions). All the features are extracted from the *test set* in CIFAR-10. A part of the eigenvalues is shown for better visualization.

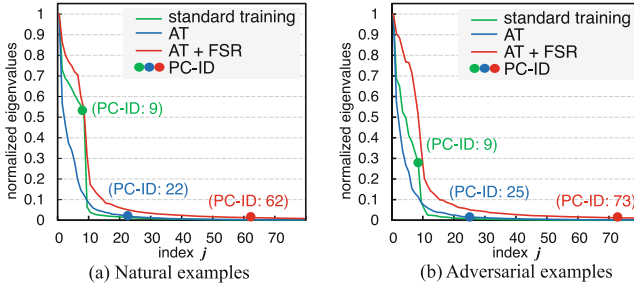


Fig. 2. Spectral analysis with features extracted from (a) natural examples and (b) adversarial examples on CIFAR-10. We scale all the eigenvalues. “PC-ID” [1] denotes the estimated intrinsic dimension (ID) of features. The sharp distribution of eigenvalues in standardly trained model leads to a lower ID, while ID becomes higher by imposing adversarial training and FSR.

Difference of models in spectral properties. As shown in Fig. 2(a)(b), the eigenvalues of a standardly trained model drop rapidly at some point, while this tendency is much alleviated by AT. The sharp distribution of eigenvalues in standard training makes just a few eigenvalues informative according to PCA, and the eigenvectors which may endow useful features are overly penalized. Consequently, model fails to recognize the change of features along eigenvectors with smaller eigenvalues. Inspired by the above observation, we propose a **hypothesis** that *the severe dominance of the top eigenvectors is a cause of vulnerability in DNN, and the adversary projects on more components in eigenvectors with smaller eigenvalues*. We will verify the proposed hypothesis in the next section.

Connection with intrinsic dimensionality. To quantitatively describe the decreasing tendency in eigenvalues, we introduce intrinsic dimensionality (ID) *i.e.*, the minimal number of parameters needed to describe a representation. ID has a close connection with natural accuracy [1], and reduction of ID contributes to an improvement on natural accuracy. We adopt PC-ID proposed by [1] to estimate ID, which is determined by the number of principal components included to describe 90% of the variance. As shown in Fig. 2, ID of a standardly trained model is very small, while models obtained by AT is higher. This also verifies that there exists a trade-off between generalization and robustness [30, 40] from the perspective of ID. Our proposed FSR could further increase ID, based on AT.

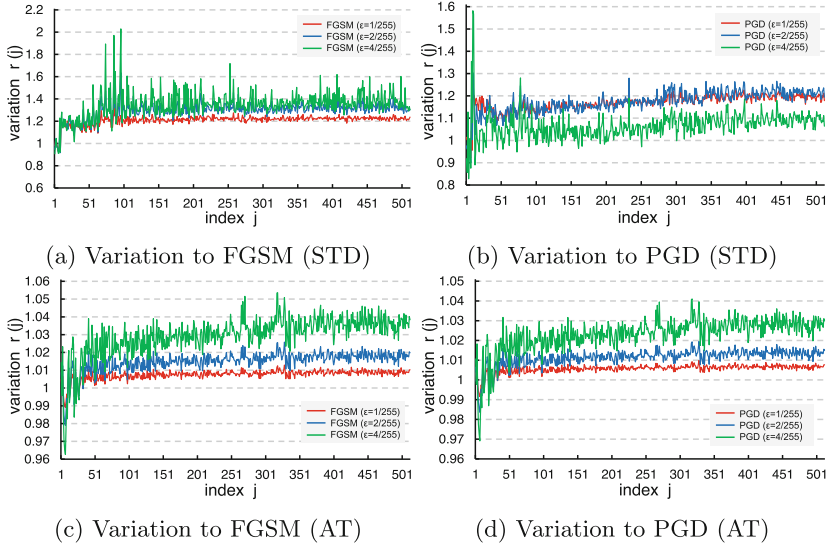


Fig. 3. Variation of all eigenvectors in feature space to adversarial attacks on CIFAR-10. “STD” means training on natural examples. “AT” means training on adversarial examples. The results reveal that the adversary projects on more components along the eigenvectors with smaller eigenvalues, and this phenomenon is alleviated by adversarial training. *It is noteworthy that the range of ordinate values for “STD” is different from “AT”.*

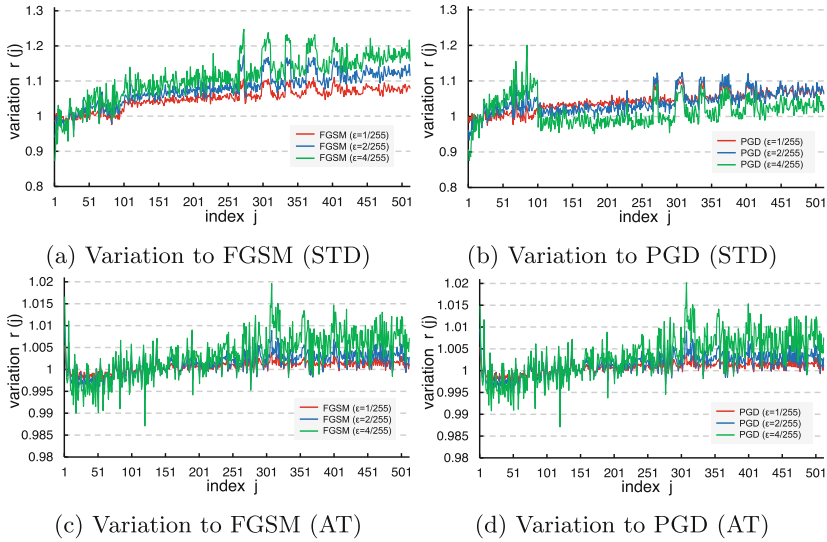


Fig. 4. Variation of eigenvectors in feature space under attack on CIFAR-100.

3.2 Variation along Eigenvectors under Attacks

To verify the hypothesis that adversary projects on more components along eigenvectors with smaller eigenvalues, we define a metric to quantitatively describe the change of features along different eigenvectors under attack, called *variation*.

Definition 1 (Alignment). Given a dataset $\mathcal{D}_s = \{\mathbf{x}_{s,i}, y_{s,i}\}_{i=1}^n$ which may be perturbed. The **alignment** of \mathcal{D}_s to the pre-given direction \mathbf{u}_j is calculated by the expectation over cosine similarity between features extracted by DNN and the direction vector \mathbf{u}_j :

$$\text{align}(\mathcal{D}_s, \mathbf{u}_j) = \mathbb{E}_{(\mathbf{x}_{s,i}, y_{s,i}) \in \mathcal{D}_s} \frac{|(h(\mathbf{x}_{s,i}), \mathbf{u}_j)|}{\|h(\mathbf{x}_{s,i})\| \cdot \|\mathbf{u}_j\|}, \quad (5)$$

where the norm $\|\cdot\|$ used is Euclidean norm, and \mathbf{u}_j is calculated by Eq. (4). The calculation of \mathbf{u}_j is based on features covariance of natural examples.

Definition 2 (Variation). Given a dataset \mathcal{D} consist of natural examples and its perturbed dataset \mathcal{D}_{adv} . The **variation** on direction \mathbf{u}_j is defined as the ratio between **alignment** on \mathcal{D}_{adv} and \mathcal{D} :

$$r(\mathcal{D}_{adv}, \mathcal{D}, \mathbf{u}_j) = \frac{\text{align}(\mathcal{D}_{adv}, \mathbf{u}_j)}{\text{align}(\mathcal{D}, \mathbf{u}_j)}. \quad (6)$$

The alignment is correlated with the distance between subspace spanned by \mathbf{u}_j and the feature space, so the change of alignment is suitable to describe the influence of attacks on direction \mathbf{u}_j . Our metric is similar to [15] in analyzing backdoors, but we define the alignment by cosine similarity while the latter uses the inner product. Compared with inner product, *cosine similarity could eliminate the influence of scale*. We give an intuitive explanation about the defined metric in Fig. 5. Suppose the distribution transfers from Fig. 5(a) to (b) under attack, we draw an original data \mathbf{x} and its shifted data \mathbf{x}_{adv} .

Take the fixed direction \mathbf{u}_2 as an illustration, the cosine similarity between \mathbf{x} and \mathbf{u}_2 is the defined *alignment*. If the distribution moves from Fig. 5(a) to (b), then alignment increases. Consequently, the change of cosine similarity can describe how the features change along a direction under attacks. For the variation defined in Eq. (6), $r(\mathcal{D}_{adv}, \mathcal{D}, \mathbf{u}_j) > 1$ means the features project on more components along direction \mathbf{u}_j , and vice versa. Therefore, we compare $r(\mathcal{D}_{adv}, \mathcal{D}, \mathbf{u}_j)$ with 1 to observe whether the adversary adds or reduces the components along the eigenvectors.

The adversary projects on more components along the eigenvectors with smaller eigenvalues. The results of visualization on CIFAR-10

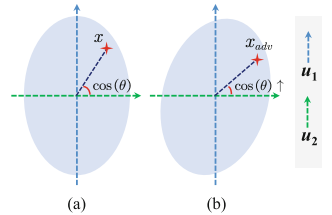


Fig. 5. A toy model that demonstrates the validity of the defined variation.

and CIFAR-100 are shown in Fig. 3~4. The attacks include FGSM [13] and PGD (step size $\epsilon/10$ for 10 steps) [21]. We set attack budget ϵ from $\frac{1}{255}/\frac{2}{255}/\frac{4}{255}$ constrained by ℓ_∞ norm. As observed in Fig. 3(a), the variation keeps close or smaller than 1 for the several largest eigenvalues in standardly trained model. However, the variation of smaller eigenvalues is much larger than 1. This indicates that FGSM tends to project on more components along the eigenvectors \mathbf{u}_j with smaller eigenvalues. A similar phenomenon also exists in other attack and datasets. For models trained by AT, variation of all eigenvectors keeps close to 1, and the high variation of directions with smaller eigenvalues visibly decreases. Similar to [17], the features along the eigenvectors with larger eigenvalues are regarded as robust features, and these along the direction with smaller eigenvalues are non-robust features. The analysis above motivates us to regularize the spectrum signatures.

4 Feature Spectral Regularization

In this section, we propose a method to regularize the distribution of eigenvalues, aiming to alleviate the dominance of the top eigenvalues. We first present our method, and then provide a theoretical analysis.

4.1 Realization of FSR

Through the analysis above, the sharp distribution of eigenvalues weakens the information contained by the smaller eigenvalues, and causes fragility to adversarial attacks. A promising approach to alleviate such phenomenon is to suppressing the largest eigenvalues, which could mitigate the dominance of the largest eigenvalues relatively. Another straightforward idea is to increase the small eigenvalues during training. However, small eigenvalues are usually numerically unstable, which may be easily affected by noise or round-off error in optimization. In this paper, we propose a method called **Feature Spectral Regularization (FSR)** by penalizing the largest eigenvalues of feature covariance in a batch of data:

$$L_{FSR}(F) = w(\tau) \cdot \lambda_{\max} \left(\left(F - \frac{1}{m} \mathbf{1}F \right)^T \left(F - \frac{1}{m} \mathbf{1}F \right) \right), \quad (7)$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a matrix, $w(\tau) = \min\{\tau/T_0, 1\}$ is the weighting function of current epoch τ with a hyper-parameter T_0 , and $\mathbf{1} \in \mathbb{R}^{m \times m}$ is a constant matrix with each element equal to 1 for calculating the mean. Moreover, $F \in \mathbb{R}^{m \times d}$ is the feature matrix of a batch composed of row vectors $h(\mathbf{x}_i)^T$, *i.e.*, $F = [h(\mathbf{x}_1) | h(\mathbf{x}_2) | \dots | h(\mathbf{x}_m)]^T$, and m is the number of samples in a batch of data.

In practice, we can just access the batch of data to approximate the statistics of feature space. However, the eigenvalues change drastically in the early stages of training, which may cause instability on optimization, so we apply a piecewise

linear function $w(\tau)$ to smooth the training stage. The $\lambda_{\max}(\cdot)$ in Eq. (7) is equal to the square of the largest singular value in feature matrix $(F - \frac{1}{m}\mathbf{1}F)$ by Singular Value Decomposition (SVD). The realization of FSR is very simple with the help of PyTorch in a few lines of code. In code written based on PyTorch, backpropagation can be automatically executed by the program.

Summarized objective loss. Adversarial training [21] has been widely proven to be a strong baseline in adversarial defense. We build and incorporate the proposed FSR into AT training framework for further improving model robustness significantly. The final objective is:

$$L_{adv}(x, y; \theta) = L_{CE}(x_{adv}, y; \theta) + \beta_{FSR} \cdot L_{FSR}(F_{adv}), \quad (8)$$

where β_{FSR} is a hyper-parameter that controls the trade-off between two items, x_{adv} represents the adversarial example generated from x by PGD [21] using cross-entropy loss, and F_{adv} is the feature matrix of adversarial examples. Besides AT, we also apply FSR to strong defense methods like TRADES [40].

Computational complexity. SVD is an extra computational cost induced by FSR. For a matrix $F \in \mathbb{R}^{m \times d}$, the time complexity of SVD is $O(\min\{m^2 \cdot d, m \cdot d^2\})$. Since the batch size is often small, the excess cost in computation is negligible compared to adversarial training.

4.2 Theoretical Analysis

Consider a linear regression model $\hat{y} = \langle z, \theta \rangle$ with ℓ_2 perturbation δ based on the feature $z \in \mathbb{R}^d$, **the underlying parameter obtained by minimizing mean square error** is $\theta_0 \in \mathbb{R}^d$ without perturbation. We assume the mean of features $\mathbb{E}(z) = 0$ and covariance matrix $\text{Var}(z) = \Sigma$. Following the adversarial risk define by [35], \mathcal{R}_{adv} is expressed in Eq. (9):

$$\mathcal{R}_{adv}(\theta, \delta) = \mathbb{E}_z \max_{\|z_{adv} - z\| \leq \delta} (\langle z_{adv}, \theta \rangle - \langle z, \theta_0 \rangle)^2. \quad (9)$$

The optimal solution of Eq. (9) denoted as θ_{adv} has the following formulation with the ridge regression:

$$\theta_{adv} = (\Sigma + \lambda I)^{-1} \Sigma \theta_0, \quad (10)$$

where λ can be regarded as a constant [35].

Given samples $\{z_i, y_i\}_{i=1}^n$, the feature matrix composed of row vector z_i^T is denoted as $Z \in \mathbb{R}^{n \times d}$, then $\Sigma = \frac{1}{n} Z^T Z$. $y = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$ is a column vector composed of real-valued output y_i ($i = 1, \dots, n$). Previous methods tend to regularize the classifier θ [36]. However, we could directly analyse how the feature Z influences the robustness of model. If the feature is more robust for classifier, then whether using the classifier θ_0 or θ_{adv} should have the same prediction. Thus, we define the *residual risk* induced by features Z :

$$\min_Z \mathcal{R}_{res}(Z) = \|Z\theta_{adv} - Z\theta_0\|_2, \quad s.t. \|Z\|_F^2 = s_0. \quad (11)$$

It is essential to normalize the scale of features for comparable representation, so we restrict the norm of Z as used in [38]. The SVD of matrix $Z \in \mathbb{R}^{n \times d}$ has the form: $Z = UDV^T$. Here $U = [\mathbf{u}_1, \dots, \mathbf{u}_n] \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{d \times d}$ are orthogonal matrices. Observe that the constraint for Z only depends on its singular values, *i.e.*, $\|Z\|_F^2 = \sum_{i=1}^{\min\{n,d\}} \sigma_i^2 = s_0$. Suppose $d < n$ and $\sigma_i > 0$, then \mathcal{R}_{res} is simplified to the expression by combining Eq. (11) and Eq. (10):

$$\min_{(\sigma_1, \dots, \sigma_d)} \mathcal{R}_{res}(\sigma_1, \dots, \sigma_d) = \min_{(\sigma_1, \dots, \sigma_d)} \left\| \sum_{j=1}^d \mathbf{u}_j \frac{\lambda n}{\sigma_j^2 + \lambda n} \mathbf{u}_j^T y \right\|_2, \text{ s.t. } \sum_{j=1}^d \sigma_j^2 = s_0. \quad (12)$$

Theorem 1. $\mathcal{R}_{res}(\sigma_1, \dots, \sigma_d)$ is minimum when all the singular values of Z are equal.

As FSR penalizes the largest singular value in feature matrix, it alleviates the dominance of large singular values and helps contribute to equal singular values under the normalized condition, helping reduce the residual risk (Table 2).

Table 1. Test accuracy (%) on CIFAR-10 under white-box attacks using ResNet-18. The maximum ℓ_∞ perturbation is $\epsilon = 8/255$. The best results are boldfaced for highlight. ‘‘Natural’’ means the classification accuracy on clean images.

Defense	Natural	FGSM	PGD-20	C&W	AA
AT	82.14±0.24	57.38±0.37	51.52±0.19	50.52±0.28	48.07±0.14
AT + FSR	82.57 ±0.36	58.02 ±0.52	52.12 ±0.16	51.36 ±0.17	48.91 ±0.17
TRADES	83.73±0.06	58.09±0.14	51.10±0.10	49.67±0.13	48.18±0.13
TRADES + FSR	84.08 ±0.48	58.43 ±0.10	51.66 ±0.05	50.00 ±0.08	48.62 ±0.17

Table 2. Test accuracy (%) on CIFAR-100 under white-box attacks using ResNet-18. The maximum ℓ_∞ perturbation is $\epsilon = 8/255$.

Defense	Natural	FGSM	PGD-20	C&W	AA
AT	55.63 ±0.06	30.87±0.12	27.62±0.20	26.14±0.18	23.93±0.17
AT + FSR	54.58±0.05	32.16 ±0.14	29.01 ±0.26	26.88 ±0.37	24.76 ±0.14
TRADES	56.03±0.65	29.84±0.07	26.22±0.25	23.83±0.32	22.81±0.20
TRADES + FSR	57.64 ±0.16	32.10 ±0.08	28.38 ±0.15	25.14 ±0.21	23.85 ±0.17

Discussion about the linear model. We present a sample statistical setting where we rigorously uncover the inherent connection between spectral properties and adversarial robustness. It is noteworthy that the theoretical analysis in the

Table 3. Test accuracy (%) on SVHN under white-box attacks using ResNet-18. The maximum ℓ_∞ perturbation is $\epsilon = 8/255$.

Defense	Natural	FGSM	PGD-20	C&W	AA
AT	90.16 \pm 0.21	59.94 \pm 0.47	47.86 \pm 0.21	45.24 \pm 0.38	42.00 \pm 0.20
AT + FSR	89.37 \pm 0.75	59.79 \pm 0.77	50.86 \pm 0.38	47.85 \pm 0.24	44.41 \pm 0.24
TRADES	92.48 \pm 0.11	68.73 \pm 0.15	58.82 \pm 0.34	55.48 \pm 0.21	52.56 \pm 0.06
TRADES + FSR	92.39 \pm 0.35	69.72 \pm 0.29	59.04 \pm 0.22	55.64 \pm 0.07	52.78 \pm 0.28

linear case is meaningful and widely used. Many theoretical works have adopted the linear case for analytical solutions since simple settings can manifest as special cases of more complex settings. For example, the work of [30] used a linear model to theoretically analyze the trade-off between robustness and accuracy.

5 Experiments

In this section, we evaluate the effectiveness of the proposed FSR on CIFAR-10 [19], CIFAR-100 [19] and SVHN. FSR is applied to two baselines: 1) AT [21, 27]; 2) TRADES [40]. It is noteworthy that AT is the most effective method to improve adversarial robustness [22, 27] in RobustBench [10].

Table 4. Test accuracy (%) based on *final checkpoint* under white-box attacks using ResNet-18 on CIFAR-10. The maximum ℓ_∞ perturbation is $\epsilon = 8/255$.

Defense	CIFAR-10		CIFAR-100		SVHN	
	PGD-20	C&W	PGD-20	C&W	PGD-20	C&W
AT	43.65	44.17	19.94	20.46	41.94	43.35
AT + FSR	44.91	44.79	22.46	21.12	44.04	44.25
TRADES	50.85	49.66	26.32	23.44	57.99	55.05
TRADES + FSR	51.15	49.54	28.43	24.68	59.11	59.13

Experimental Settings. For CIFAR-10 and CIFAR-100, we set the ℓ_∞ perturbation with $\epsilon = 8/255$, the step size of attack $2/255$, and the inner iteration steps 10. The step size is $1/255$ for SVHN. We train ResNet-18 [16] using momentum optimizer with the initial learning rate of 0.1. The weight decay factor is set to $5e - 4$. For AT, we train 200 epochs and the learning rate decays with a factor of 0.1 at 100 and 150 epochs [27]. For TRADES, we train 120 epochs with the learning rate divided by 0.1 at epochs 75, 90, and 100 [40]. The parameter for regularization ($1/\lambda$) is set as 4 for TRADES. For FSR, we set $\beta_{FSR} = 0.01$. Other hyper-parameters keep the same as their original paper. Considering that the settings have a distinct influence on robustness [22], the hyper-parameters remain unchanged while adding our FSR.

5.1 Performance under white-box attacks

We adopt various white-box adversarial attacks: FGSM [13], PGD-20 (step size $\epsilon/4$) [21], C&W (ℓ_∞ version optimized by PGD) [7]. Following the instruction proposed in [6], we use 5 random starts at random offsets away from the initial when applying iterative attacks. The settings for evaluation promote a quite strong attack. We report the best checkpoint (the highest robustness under PGD from different checkpoints) as used in [27]. The test accuracy is reported in Table 1~3. The results show that FSR improves adversarial robustness. We also test the robustness under AutoAttack (AA) [11], which is now regarded as the strongest attack. FSR is also effective to improve robustness under AA. The detailed results show that FSR indeed boosts robustness, rather than depending on obfuscated gradients [2]. The improvement on various datasets reveals that FSR has a consistent promotion. We also report the performance of final checkpoint in Table 4, showing that FSR improves the performance in final checkpoint.

Combination with other methods. To validate that FSR is an effective module, we combine FSR with Adversarial Weight Perturbation (AWP) [33], which is one of the strongest defense in RobustBench [10]. The results are listed in Table 5, showing that FSR *attains improvement on adversarial robustness based on AWP*, especially under CW attack and AA.

Performance under WideResNet. We conduct experiments on the larger network WideResNet-34-10 [39]. We realize our method based on AT [27] and AT-AWP [33], as shown in Table 6. The “Best” means the *checkpoint of best performance under PGD*, and the “Last” is the performance of final checkpoint. The results verify that our method is also effective in WideResNet. Experiments based on some new architectures (such as vision transformers [12]) require many detailed modifications for adversarial training [4]. The related theoretical analysis and experiments are worth exploring as a future direction.

Remark. We would like to provide supplementary explanations for these results. (1) The paper’s experiments cover a broad spectrum, including various attacks, defenses, backbones, and datasets. These consistently and steadily validate the effectiveness of FSR. (2) We repeat the experiments in Table 1~3. The comprehensive results reveal a steady improvement. (3) The proposed FSR is a simple module readily pluggable into any defense method in a few lines of codes. FSR could achieve a competitive improvement with recent work [18, 25] on AutoAttack. In addition, FSR is motivated by the connection between spectral signatures and robustness, so it could provide new insights to understand adversarial robustness.

5.2 Performance under black-box attacks

Transferability is an intriguing property for adversarial example to implement black-box attack [23]. Transfer attack is beneficial to verify that our method does not rely on gradient masking. As suggested by [6], we adopt another adversarially

Table 5. Test accuracy (%) on CIFAR-10 under white-box attacks using ResNet-18 based on AWP. We report the results based on best checkpoint. The maximum ℓ_∞ perturbation is $\epsilon = 8/255$.

Defense	Natural FGSM PGD-20 C&W AA				
AT-AWP	80.23	59.03	55.13	51.38	49.48
AT-AWP + FSR	80.63	59.11	55.13	51.99	49.93

Table 6. Test accuracy (%) on CIFAR-10 under white-box attacks using WideResNet-34-10.

Defense	FGSM		PGD-20		C&W	
	Best	Last	Best	Last	Best	Last
AT	61.85	57.69	55.10	47.46	53.54	48.12
AT + FSR	62.22	59.12	54.93	47.89	53.98	48.32
AT-AWP	63.64	63.96	58.17	57.09	55.81	55.11
AT-AWP + FSR	64.19	63.59	58.23	58.01	55.99	56.25

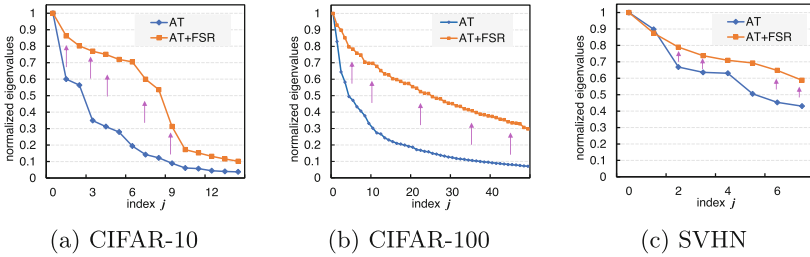


Fig. 6. Spectral analysis of FSR. (a) normalized eigenvalues on CIFAR-10; (b) normalized eigenvalues on CIFAR-100; (c) normalized eigenvalues on SVHN.

trained ResNet-18 as the substitute model. The results are listed in Table 7, revealing that FSR also improves robustness under black-box attacks.

Table 7. Black-box attack robustness (%) under transfer attack on CIFAR-10, CIFAR-100 and SVHN. The parameters of threat model are the same as white-box attack. The best results are boldfaced for highlight. The backbone is ResNet-18.

Defense	CIFAR-10			CIFAR-100			SVHN		
	FGSM	PGD	C&W	FGSM	PGD	C&W	FGSM	PGD	C&W
AT	63.63	61.12	61.36	41.50	40.94	42.27	67.24	62.09	63.57
AT + FSR	63.74	61.25	61.63	41.93	41.42	42.69	67.39	62.72	64.54
TRADES	65.34	62.94	62.68	40.76	39.98	41.31	73.13	68.90	70.11
TRADES + FSR	65.98	63.60	63.40	42.57	41.91	43.22	73.35	69.32	70.60

Table 8. Test accuracy (%) under white-box attacks for different β_{FSR} on CIFAR-10. The backbone is ResNet-18.

β_{FSR}	0	0.005	0.010	0.020	0.060
Natural	82.02	81.96	82.18	81.84	81.74
PGD-20	51.63	52.42	52.24	52.77	52.21
C&W	50.28	51.04	50.77	51.21	50.75

Table 9. Test accuracy (%) on CIFAR-10 under AutoAttack using ResNet-18 while suppressing the largest k eigenvalues.

k	1	2	4	8	12	16
AA	48.71	48.95	49.02	49.17	48.55	48.14

Besides, we analyse the influence of FSR on the eigenvalue spectrum, as shown in Fig. 6. To eliminate the influence from the scale of features, the eigenvalues are divided by the maximum eigenvalue. As shown in the figure, FSR increases the eigenvalues relatively, which is consistent with our intention for FSR.

5.3 Ablation Studies

Sensitivity analysis of β_{FSR} . We explore how the weight of FSR β_{FSR} influences the performance, as listed in Table 8. It reveals that FSR significantly improve robustness with a wide value range. In this paper, we choose $\beta = 0.01$ considering both generalization and robustness.

Suppressing more eigenvalues. In previous part, FSR only penalizes the largest eigenvalue. We explore the influence of penalizing more eigenvalues in Table 9. As properly increasing the number, the robustness is further improved. However, if we further keep increasing k , the robustness declines. We think it is due to that k has some trade-off with the weight of FSR.

6 Conclusion

In this paper, we delve into the discrepancy between natural examples and adversarial examples from the perspective of spectral analysis. The variation of different eigenvectors under adversarial attacks is analysed. It is shown that the spectral directions with smaller eigenvalues are more fragile under attack, which is induced by the dominance of the top eigenvectors. Based on the analysis, a method called Feature Spectral Regularization (FSR) is proposed to penalize the largest eigenvalues of the batch covariance matrix, which is numerically stable and could enlarge the overall eigenvalues relatively. We also provide a theoretical analysis in robust linear regression. FSR is a simple module readily pluggable

into any defense method. Through comprehensive experiments, we show that FSR can effectively improve adversarial robustness.

Acknowledgments. This work has been supported by the National Science and Technology Major Project (2022ZD0116500), and National Natural Science Foundation of China (62222609, 62076236).

References



1. Ansuini, A., Laio, A., Macke, J.H., Zoccolan, D.: Intrinsic dimension of data representations in deep neural networks. In: *Advances in Neural Information Processing Systems*. vol. 32 (2019)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: *International Conference on Machine Learning*. pp. 274–283. PMLR (2018)
3. Bai, Y., Zeng, Y., Jiang, Y., Xia, S.T., Ma, X., Wang, Y.: Improving adversarial robustness via channel-wise activation suppressing. In: *International Conference on Learning Representations* (2021)
4. Bai, Y., Mei, J., Yuille, A.L., Xie, C.: Are transformers more robust than cnns? In: *Advances in Neural Information Processing Systems*. vol. 34, pp. 26831–26843 (2021)
5. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 387–402. Springer (2013)
6. Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. arXiv preprint [arXiv:1902.06705](https://arxiv.org/abs/1902.06705) (2019)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: *2017 IEEE Symposium on Security and Privacy (SP)*. pp. 39–57. IEEE (2017)
8. Carmon, Y., Raghuathan, A., Schmidt, L., Duchi, J.: Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems* (2019)
9. Chen, X., Wang, S., Long, M., Wang, J.: Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In: *International Conference on Machine Learning*. pp. 1081–1090. PMLR (2019)
10. Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. arXiv preprint [arXiv:2010.09670](https://arxiv.org/abs/2010.09670) (2020)
11. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: *International Conference on Machine Learning*. pp. 2206–2216. PMLR (2020)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021)
13. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: *International Conference on Learning Representations* (2015)
14. Goyal, S., Rebuffi, S.A., Wiles, O., Stimberg, F., Calian, D.A., Mann, T.A.: Improving robustness using generated data. In: *Advances in Neural Information Processing Systems*. vol. 34, pp. 4218–4233 (2021)

15. Hayase, J., Kong, W., Somani, R., Oh, S.: Defense against backdoor attacks via robust covariance estimation. In: International Conference on Machine Learning. pp. 4129–4139. PMLR (2021)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
17. Ilyas, A., Santurkar, S., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
18. Kanai, S., Yamaguchi, S., Yamada, M., Takahashi, H., Ohno, K., Ida, Y.: One-vs-the-rest loss to focus on important samples in adversarial training. In: International Conference on Machine Learning. pp. 15669–15695. PMLR (2023)
19. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images (2009)
20. Lezama, J., Qiu, Q., Musé, P., Sapiro, G.: Ole: Orthogonal low-rank embedding—a plug and play geometric loss for deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8109–8118 (2018)
21. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
22. Pang, T., Yang, X., Dong, Y., Su, H., Zhu, J.: Bag of tricks for adversarial training. In: International Conference on Learning Representations (2021)
23. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint [arXiv:1605.07277](https://arxiv.org/abs/1605.07277) (2016)
24. Papayan, V., Han, X., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proc. Natl. Acad. Sci. **117**(40), 24652–24663 (2020)
25. Rade, R., Moosavi-Dezfooli, S.M.: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In: International Conference on Learning Representations (2022)
26. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
27. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning. pp. 8093–8104. PMLR (2020)
28. Roth, K., Kilcher, Y., Hofmann, T.: Adversarial training is a form of data-dependent operator norm regularization. In: Advances in Neural Information Processing Systems. vol. 33, pp. 14973–14985 (2020)
29. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
30. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2019)
31. Wang, Z., Grigsby, J., Qi, Y.: PGrad: Learning principal gradients for domain generalization. In: International Conference on Learning Representations (2023)
32. Wei, Z., Wang, Y., Guo, Y., Wang, Y.: Cfa: Class-wise calibrated fair adversarial training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8193–8201 (2023)

33. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. In: *Advances in Neural Information Processing Systems*. vol. 33, pp. 2958–2969 (2020)
34. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 501–509 (2019)
35. Xing, Y., Zhang, R., Cheng, G.: Adversarially robust estimate and risk analysis in linear regression. In: *International Conference on Artificial Intelligence and Statistics*. pp. 514–522. PMLR (2021)
36. Xu, H., Caramanis, C., Mannor, S.: Robust regression and lasso. In: *Advances in Neural Information Processing Systems*. vol. 21 (2008)
37. Yan, H., Zhang, J., Niu, G., Feng, J., Tan, V., Sugiyama, M.: Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In: *International Conference on Machine Learning*. pp. 11693–11703. PMLR (2021)
38. Yu, Y., Chan, K.H.R., You, C., Song, C., Ma, Y.: Learning diverse and discriminative representations via the principle of maximal coding rate reduction. *Advances in Neural Information Processing Systems* **33** (2020)
39. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: *British Machine Vision Conference 2016*. British Machine Vision Association (2016)
40. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: *International Conference on Machine Learning*. pp. 7472–7482. PMLR (2019)



Cluster-Mined Negative Samples for Enhanced Unsupervised Sentence Representation Learning

Yuhang Zhang¹ , Wenjie Zhang¹, Yang Hua¹, Zun Wang¹,
Xiaoning Song^{1,2} , and Xiao-jun Wu¹

¹ School of Artificial Intelligence and Computer Science, Jiangnan University,
Wuxi 214122, China
{6223114030, wenjie.zhang, 7211905018, 6223111064}@stu.jiangnan.edu.cn,
{x.song, wu_xiaojun}@jiangnan.edu.cn

² DiTu (Suzhou) Biotechnology Co., Ltd., Suzhou 215000, China

Abstract. Negative samples selection for contrastive learning is considerable in the field of sentence representation, especially for semantic textual similarity. Traditional in-batch negative sampling methods not only lack hard negative samples but also ignore potential false negative samples. Despite numerous methods trying to improve traditional sampling strategies, the challenge of consistently generating high-quality negative samples remains untackled. To address this pivotal issue, we propose the Cluster-Mined Negative Samples for Enhanced Unsupervised Sentence Representation Learning (CMNS) framework. Specifically, dynamic queues are utilized to store the K-means cluster samples, enabling the most appropriate selection of clusters to serve as negative samples. Additionally, we generate noise-based negative samples via stored clusters while simultaneously constraining potential false negative samples. Above all, CMNS employs clustering techniques to efficiently mine sufficient quantity of high-quality negative samples from unlabeled datasets. Extensive experiments illustrate that our approach not only overcomes the inherent limitations of traditional sampling methods but also improves the performance of sentence representations in downstream tasks, demonstrating measurable advancements over current methodologies. (codes and models are available at <https://github.com/hamrain/CMNS>).

Keywords: Sentence Representation · Contrastive Learning · Clustering · Negative Sampling · Semantic Text Similarity

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78395-1_29.

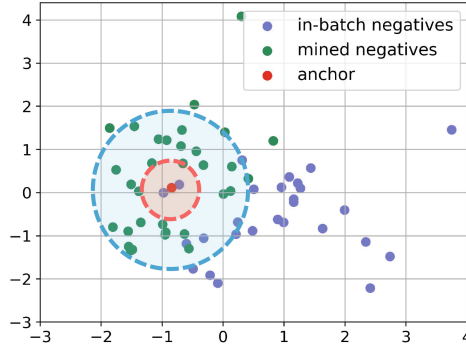


Fig. 1. We perform PCA dimensionality reduction mapping on an anchor sample (a given sentence), negative samples in the same batch, and negative samples mined by our method to a 2D plane. The red circle indicates close proximity to the anchor, while the blue circle signifies similarity with a distinct difference.

1 Introduction

Learning effective sentence representations remains a crucial task in the field of Natural Language Processing (NLP), as it lays the foundation for various downstream applications such as Semantic Textual Similarity [30], Information Retrieval [20] and Question Answering [26]. Recent advances in pre-trained language models (PLM), such as BERT [12] and RoBERTa [24], have significantly improved the quality of sentence embeddings. However, the issue of anisotropy in these embeddings persists [16, 23], thus limiting their representational capacity.

Contrastive learning has emerged as a promising approach to improve sentence representational capacity in fine-tuning PLM. By distinguishing between semantically similar and dissimilar sentence pairs, it refines sentence representations [17, 44]. Contrastive learning typically involves comparing positive and negative samples with an anchor sentence. Positive samples should be semantically close to the anchor, while negative samples should be diverse, including hard negatives that are partially similar but not identical to the anchor [15, 31]. Compared to the sustained focus on the construction of positive samples [17, 40, 43], research on negative sample selection has not received equal attention. This discrepancy becomes particularly evident during negative sampling, as researchers often simply consider other samples within the same batch as negatives.

We can observe some shortcomings of in-batch samples from Figure 1. Firstly, the distribution of distances between these in-batch negative samples and the anchor point is overly random, with some even overlapping the anchor point, and they tend to cluster densely in the lower right corner of the plane. Negative samples selected in-batch are likely to include false negatives, which are sentences that are semantically close to the anchor but are incorrectly labeled as dissimilar. This negative sampling method can lead to the model mistakenly pushing away semantically related sentences, thereby hindering the learning process. Moreover,

due to the anisotropy problem, the representations of the selected negative samples often originate from a narrow representation cone, which fails to adequately reflect the global semantics of the representation space. These issues all point to the inadequacies of the current negative sampling strategy.

Recent studies have attempted to address these issues, DCLR [44] use Gaussian distribution to generate samples, filtering those far from anchors with an auxiliary model. ClusterNS [11] use mini-batch clustering for negative sampling. Despite adopting innovative methods to reduce false negatives, neither approach can ensure the acquisition of a sufficient number of high-quality hard negative samples. Although effective negative samples can be acquired by retrieval [38] or data augmentation [35], it is still a time-consuming process. Thus, getting quality hard negatives remains a challenge in contrastive learning.

In order to address the restrictions of existing contrastive learning approaches in negative samples selection, we introduce a framework called Cluster-Mined Negative Samples for Enhanced Unsupervised Sentence Representation Learning (CMNS). This framework uses K-means clustering algorithms to mine high-quality negative samples from unlabeled datasets, thereby optimizing the contrastive learning process. Our approach primarily focuses on the selection of hard negative samples and the handling of false negative samples. First, we introduce Dynamic Clustering Queues to store and retrieve sentence representations, ensuring sample diversity and quality. Using clustering, we group samples by their cosine similarity with the clustering queues and assign each to the most similar queue. Subsequently, we choose the second most similar cluster to the anchor sample as the source of hard negatives. As shown in Figure 1, our mined negatives surround the anchor point at an appropriate distance, providing semantic information for the model sufficiently. Since samples within the same cluster lack explicit label data, we treat them as false negatives and constrain their impact using a bidirectional margin loss. Finally, we generate virtual negatives by adding Gaussian noise to the centroid of the hard negatives cluster in order to enhance the uniformity of the representation space.

Overall, our negative sampling approach provides an effective solution that seamlessly integrates with existing methods. Precisely, our approach outperforms SimCSE by 2.09% / 1.70% on BERT_{base} / RoBERTa_{base} respectively, and also surpasses PromptBERT by 0.58% on BERT_{base}. The following are our primary contributions:

1. We introduce Dynamic Clustering Queues to maintain a group of sample clusters, which enables the identification of hard negative samples and the resolution of false negative issues. Our clustering analysis indicates enhanced model discriminability.
2. We generate virtual negative samples by introducing noise to the center of hard negative samples, ensuring the diversity and completeness of the negative samples.
3. Experiments conducted on semantic textual similarity (STS) tasks show that our approach significantly outperforms baseline models.

2 Related Work

2.1 Contrastive Learning.

Contrastive learning initially achieves remarkable results in the fields of computer vision [18] and information retrieval [6]. Subsequently, Chen et al. [9] propelled contrastive learning into the mainstream by modifying the contrastive loss and introducing data augmentation techniques [9]. In the field of unsupervised sentence representation learning, utilizing data augmentation methods to generate positive pairs has also yielded significant achievements [6, 17, 23]. For instance, Gao et al. [17] employ dropout as a data augmentation technique, significantly improving performance on semantic textual similarity tasks. During this phase, the selection of negative samples primarily relied on random sampling within a batch or from the entire dataset. Subsequently, negative samples selection methods are optimized. Zeng et al. [41] attempt to derive negative templates from the negation of different prompt templates, but such fixed templates may introduce bias. Deng et al. [11] perform clustering sampling on samples within a batch, yet this approach may lack sufficiently effective negative samples. Zhou et al. [44] achieve uniformity in negative samples by generating them using random Gaussian noise and introduced an additional model to assist training, reducing the impact of false negative samples on model training. However, this method cannot fully guarantee the high-quality of negative samples.

2.2 Clustering Integration

Clustering methods are integrated into deep learning frameworks and employed for unsupervised representation learning [23, 42]. Additionally, Prototypical Networks [32], a specific clustering approach, has gained popularity in few-shot learning [13]. Furthermore, several research efforts have combined clustering with contrastive learning [11, 36]. Among these, Wang et al. [36] present a contrastive approach to clarify ambiguous labels in partial label learning, while Deng et al. [11] attempt to address the issue of negative samples quality through in-batch clustering.

3 Preliminaries

Our goal is to unsupervisedly fine-tune a pre-trained language model to enhance its sentence representation capabilities. Through fine-tuning, we expect semantically similar sentences to be closer in the embedding representation space, while sentences with significant semantic differences are dispersed farther apart. Contrastive learning has proven effective in distinguishing semantic differences between sentences. It often uses the InfoNCE loss [27] to measure similarity between positive and negative sample pairs for representation learning. For a given anchor sentence x_i , SimCSE uses dropout, and PromptBERT uses prompt templates for data augmentation, both creating positive sample x_i^+ . These methods typically choose other sentences from the same batch as negative samples

$\{x_j\}$, and the sentence embeddings presentation are then utilized in the InfoNCE loss:

$$\mathcal{L}_{InfoNCE} = -\log \frac{e^{\text{sim}(x_i, x_i^+) / \tau}}{\sum_{j=1}^N e^{\text{sim}(x_i, x_j) / \tau}}, \quad (1)$$

where $\text{sim}(x_i, x_j)$ is the cosine similarity $\frac{x_i^T \cdot x_j}{\|x_i\| \|x_j\|}$, N is the batch size and a temperature parameter τ that controls the distribution of smoothing.

Although contrastive learning aids in understanding semantic differences, the quality of negatives drawn from the same batch can limit its effectiveness, which emphasizing the importance of mining high-quality negative samples.

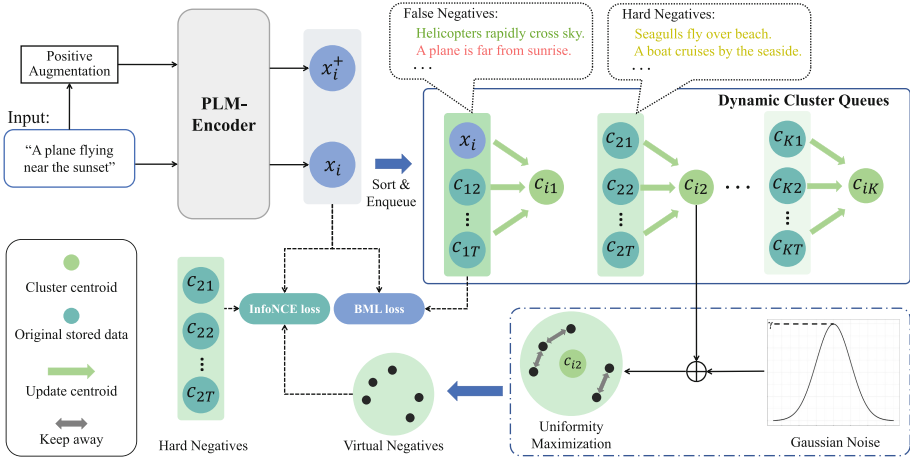


Fig. 2. Our framework CMNS uses the Dynamic Cluster Queues to create sample groups. The x_i 's second-closest cluster is treated as hard negatives and noise-based virtual negatives are generated from its centroid. Then we use InfoNCE loss to discern positives from these negatives. Additionally, we regard x_i 's cluster as false negatives, constrained by BML loss.

4 Approach

4.1 Dynamic Cluster Queues

In order to mine more high-quality negative samples, we introduce a Dynamic Clustering Queues, denoted as Q . The Q is a container of K clusters $C = \{C_k\}_{k=1}^K$, each containing T unique sentence representations. For the k -th cluster C_k , which is implemented by a queue, is used to organize similar representations of sentences. By constructing the Dynamic Clustering Queues Q , sentence representations can be stored and updated dynamically.

When initialising Q , we heuristically select the K sentence representations with the least similarity within the batch, and assign each sentence representation x_k to the cluster C_k as the initial centroid c_k . This ensures dispersion in the initial K cluster vector space. The sentence clustering process based on the K-means algorithm after initialization is as follows:

(1) For each sentence x_i in a mini-batch, we compute its cosine similarity with each cluster centroid c_k . Then, we sort the clusters in descending order of the computed similarity scores, generating an ordered list $(C_{i1}, C_{i2}, \dots, C_{iK})$. For instance, C_{i1} represents the cluster most similar to the sentence x_i , C_{i2} is the second most similar.

(2) During enqueue, we assign the sentence x_i to the cluster C_{i1} that is most similar to x_i . If the number of sentence represent in cluster C_{i1} has reached the limit T , we perform a dequeue operation. Then, we update the centroid c_{i1} of cluster C_{i1} based on the average representation of all sentences in C_{i1} .

4.2 Hard Negatives

In contrastive learning, the selection of suitable negative samples is critical for improving the performance of the model. Hard negatives [7, 18], which are highly similar to the anchor but belong to a different class, play an important role in improving the generalization capability. The Dynamic Cluster Queues effectively identify these valuable hard negatives. Specifically, we determine the cluster C_{i2} that is the second most similar to the anchor x_i , and then select all samples from cluster C_{i2} as hard negatives. These carefully chosen samples, which share semantic similarities with the anchor while maintaining fine-grained differences, provide essential gradient information for refining the model’s ability to detect subtle distinctions. Through this strategy, we ensure that the model learns more precise feature representations, ultimately enhancing its overall performance.

4.3 Virtual Negatives

The diversity of negative samples is key to effective contrastive learning, enhancing the quality of learned representations, particularly with hard negatives. To boost this diversity, we create challenging virtual negatives by adding Gaussian noise [44] to existing hard negatives. In our framework, c_{i2} represents the centroid of the cluster C_{i2} , which consists of hard negatives for x_i . Using c_{i2} as a foundation, we can generate a set of R virtual negatives Z_i by introducing Gaussian noise. These virtual negatives are designed to be both similar to and distinct from x_i :

$$Z_i = z_{ij} = c_{i2} + \gamma \mathcal{N}_j, \quad j \in \{1, 2, 3, \dots, R\}, \quad (2)$$

where γ is a hyperparameter controlling noise degree, \mathcal{N} is a random noise vector from a standard Gaussian distribution and R is the number of virtual samples to be generated.

We consider that virtual negative samples may be overly concentrated or overlapping, resulting in repetitive learning content. Therefore, we design a uniformity maximization loss (UnifMax): we calculate the minimum cosine distance between each pair of virtual negative samples, take the distances average as the regularization term, to encourage the model to learn a more uniformly distributed embedding representation. Simultaneously, the cosine similarities between virtual negatives and anchor x_i is incorporated into the loss function to avoid virtual negatives being too close to x_i in feature space. Through S-step iterative optimization as:

$$\text{UnifMax} = -\frac{2}{R(R-1)} \sum_{p=1}^R \sum_{q=p+1}^R (1 - \text{sim}(z_{ip}, z_{iq})) - \frac{1}{R} \sum_{j=1}^R \text{sim}(x_i, z_{ij}), \quad (3)$$

where z_{ip} and z_{iq} represent different virtual negatives in Z_i . In each iteration, the loss is computed using current virtual samples, which are then regenerated after each update. In this method, noise-based virtual negatives are optimised, thereby enhancing the diversity of hard negatives. Using the virtual negative generation method, we merge it with the hard negatives approach described in subsection 4.2 to produce the negatives incorporated in the InfoNCE loss:

$$\mathcal{L}_{cl} = -\log \frac{e^{\text{sim}(x_i, x_i^+)/\tau}}{\sum_j e^{\text{sim}(x_i, x_j)/\tau} + \mu \sum_{x^-} e^{\text{sim}(x_i, x^-)/\tau}}, \quad (4)$$

where x_j is the in-batch negative, $x^- \in C_{i2} \cup Z_i$ represents all negative samples generated by our framework and μ is the weight of them.

4.4 False Negatives and BML Loss

In our framework, sentences in cluster C_{i1} exhibit high similarity to the anchor sentence x_i . However, in an unsupervised learning context, capturing precise semantic relationships between sentences is challenging. As exemplified in Figure 2, while the textual similarity between sentences in C_{i1} and x_i is evident, their semantic content exhibits a degree of distinction. Therefore, we refrain from simply classifying C_{i1} as positives and instead consider them as false negatives in the semantic sense. Critically, the dynamic updating of cluster centroids continuously modulates the similarity between these false negatives and the cluster centroids, further complicating their relationship with x_i . To tackle this complexity, we adapted the bidirectional margin loss (BML) [35] for our clustering methodology. Our goal is to use BML loss to precisely differentiate between false negatives, positive pairs, and hard negatives.

Specifically We've defined two metrics: Δ_1 compares the cosine similarity of false negatives and anchor x_i to that of positive pairs. Δ_2 compares the cosine similarity between cluster C_{i2} 's centroid c_{i2} and x_i with that of false negatives and x_i :

$$\begin{aligned}
\Delta_1 &= \cos(C_{i1}, x_i) - \cos(x_i^+, x_i), \\
\Delta_2 &= \cos(c_{i2}, x_i) - \cos(C_{i1}, x_i), \\
\mathcal{L}_{bml} &= \text{ReLU}(\Delta_1 + \alpha) + \text{ReLU}(\Delta_2 + \beta),
\end{aligned} \tag{5}$$

Through the application of through the BML loss, we constrain the similarity between false negatives and x_i to lie within a specific range relative to the positive examples and the centroid c_{i2} of hard negatives. This range is defined as $[\cos(c_{i2}, x_i) + \beta, \cos(x_i^+, x_i) - \alpha]$, allowing the model to more accurately disentangle textual similarity from semantic similarity.

Finally, by combining Eq. 4 and Eq. 5, we arrive at our training objective: a weighted blend of contrastive learning loss and bidirectional margin loss:

$$\mathcal{L} = \mathcal{L}_{cl} + \lambda \mathcal{L}_{bml}, \tag{6}$$

where λ is a hyperparameter. The complete formulaic process of the CMNS is shown in Appendix A.

5 Experiments

5.1 Experiment Setup

Drawing upon previous methods [17, 22], we utilize the SentEval Toolkit [10] to conduct our experiments on seven STS tasks [1–5, 8, 25]. For evaluation, we adopt Spearman’s correlation coefficient as the metric and follow SimCSE’s aggregation methods of results method [17].

5.2 Implementation Details.

Our work is built upon the Huggingface Transformers library. The experiments are conducted on four NVIDIA RTX 2080Ti 11GB GPUs. Our models derive from SimCSE [17] and PromptBERT [21], with the former serving as the foundation for our *Non-Prompt* CMNS and the latter as the basis for our *Prompt-based* CMNS. We utilize the pre-trained BERT and RoBERTa models, which are subsequently fine-tuned on a subset of 1 million randomly selected sentences from Wikipedia. Models are trained for 1 epoch with temperature $\tau = 0.05$ using an Adam optimizer, with a learning rate adjusted according to the model size. To maintain fairness, we evaluate performance on the STS-B and SICK-R development sets at 125-step intervals throughout the training process to select the optimal checkpoint for final evaluation. Our experimental setup ensures reproducibility. Hyperparameter settings and more training details are listed in Appendix B.

Table 1. Performance evaluation of sentence embeddings on STS tasks using Spearman’s correlation coefficient. All reference outcomes are sourced from original or associated research papers. The highest scores are emphasized in bold.

Models	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
<i>Non-Prompt models</i>								
BERT _{base} (avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
ConSERT-BERT _{base}	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT _{base}	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
DCLR-BERT _{base}	70.81	83.73	75.11	82.56	78.44	78.31	71.59	77.22
ClusterNS-BERT _{base}	69.93	83.57	76.00	82.44	80.01	78.85	72.03	77.55
CMNS-BERT _{base}	73.10	83.98	76.57	83.19	80.06	79.45	72.00	78.34
RoBERTa _{base} (avg.)	32.11	56.33	45.22	61.34	61.98	54.53	62.03	53.36
SimCSE-RoBERTa _{base}	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
IS-CSE-RoBERTa _{base}	71.39	82.58	74.36	82.75	81.61	81.40	69.99	77.73
DCLR-RoBERTa _{base}	70.01	83.08	75.09	83.66	81.06	81.86	70.33	77.87
CMNS-RoBERTa _{base}	73.29	83.40	75.00	82.64	82.00	81.77	69.82	78.27
<i>Prompt-based models</i>								
PromptBERT _{base}	71.56	84.58	76.98	84.47	80.60	81.60	69.87	78.54
ClusterNS-BERT _{base}	72.92	84.86	77.38	84.52	80.23	81.58	69.53	78.72
ConPVP-BERT _{base}	71.72	84.95	77.68	83.64	79.76	80.82	73.38	78.85
SNCSE-BERT _{base}	70.67	84.79	76.99	83.69	80.51	81.35	74.77	78.97
CMNS-BERT _{base}	73.48	85.76	77.20	83.39	80.91	81.96	71.11	79.12

5.3 Main Results

We present the experimental results across seven STS tasks in Table 1, comparing various sentence embedding models: 1) Baseline models: SimCSE [17] and PromptBERT [21]. 2) Variations of SimCSE models: DCLR [44], ClusterNS [11] and IS-CSE [43]. 3) Extensions of PromptBERT models ConPVP [41] and SNCSE [35]. These models are experimented on BERT and RoBERTa separately. To conduct a comprehensive and fair comparison with the two baseline models, as well as their variants, we designed experiments specifically targeting *Non-Prompt* CMNS and *Prompt-based* CMNS, respectively.

The following are our main conclusions: when compared to the two baseline models, SimCSE and PromptBERT, the CMNS models demonstrate improvements of 2.09% and 0.58% on BERT, indicating the effectiveness and importance of negative sampling. In the case of non-prompt models, CMNS surpasses models like DCLR and ClusterNS. For prompt-based models, CMNS exceeds SNCSE and ConPVP on BERT. All these models enhance negative sampling

through various sampling or construction methods, highlighting the excellent performance of our model.

5.4 Ablation Study

Our proposed framework encompasses three key modules: mining hard negatives from original samples, creating virtual negatives via Gaussian noise, and addressing false negatives. To thoroughly investigate the contributions of these modules and their synergistic effects, we undertook comprehensive ablation studies on the STS tasks. During the experimentation phase, we leveraged *Non-prompt* BERT and RoBERTa to assess each component individually and in various combinations. Furthermore, to confirm the efficacy of our designed approach, we made modifications to certain components. Firstly, we evaluated our hard negatives selection by comparing two approaches: using the cluster C_{i1} most similar to anchor x_i (named *repl. FirstCluster*) as hard negatives, and using a randomly selected cluster (named *repl. RandCluster*) as hard negatives. Additionally, we bypassed Uniformity-based optimization and directly generated negative samples using random noise derived from c_{i2} (named *BypassUniform*).

Table 2. Ablation outcomes for our *Non-prompt* Models on the STS task test dataset.

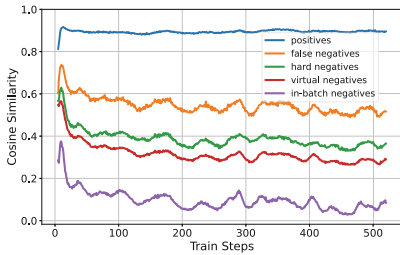
Models	BERT _{base}	RoBERTa _{base}
CMNS	78.34	78.27
<i>w/o hard negs</i>	77.75 _{↓0.59}	77.50 _{↓0.77}
<i>w/o virtual negs</i>	77.82 _{↓0.52}	77.81 _{↓0.46}
<i>w/o BML loss</i>	77.91 _{↓0.43}	77.88 _{↓0.39}
<i>only hard negs</i>	77.56 _{↓0.78}	77.58 _{↓0.69}
<i>only virtual negs</i>	77.48 _{↓0.86}	77.39 _{↓0.88}
<i>only BML loss</i>	76.02 _{↓2.32}	76.43 _{↓1.84}
<i>repl. FirstCluster</i>	75.16 _{↓3.18}	74.91 _{↓3.36}
<i>repl. RandCluster</i>	77.39 _{↓0.95}	77.65 _{↓0.62}
<i>BypassUniform</i>	77.48 _{↓0.86}	77.67 _{↓0.60}
SimCSE	76.25	76.57

Table 2 shows that deleting or replacing CMNS components leads to reduced performance compared to the original setup, underlining the significance of each component within the CMNS structure: 1) Providing more negative samples resulted in further improvements, as we employed clustering to generate samples with greater similarity. 2) Using BML loss alone to handle false negative samples may actually decrease performance, implying that the model tends to

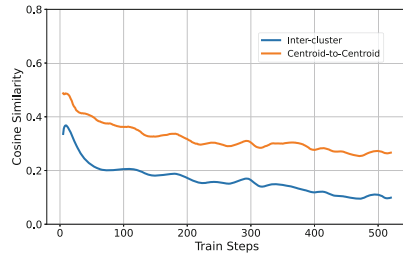
learn more from hard negatives than from correctly identifying true false negatives. 3) The effectiveness of using the first or random cluster as hard negatives is far inferior to that of the second cluster. This observation proves the importance of selecting samples that are similar but not excessively close as negative samples for the model, thereby confirming the correctness of our selection of the second cluster. 4) The lack of a uniformity process between negative samples leads to performance degradation, indicating that simply using random Gaussian noise may cause information overlap.

6 Analysis

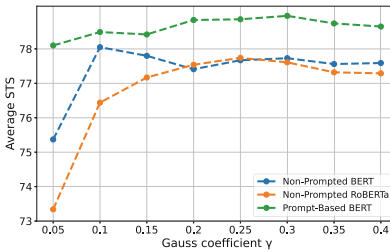
To better comprehend clustering’s role in training, we visualize key information fluctuations during the process in the *Non-Prompt* CMNS-BERT_{base} model and conduct a detailed analysis of the results.



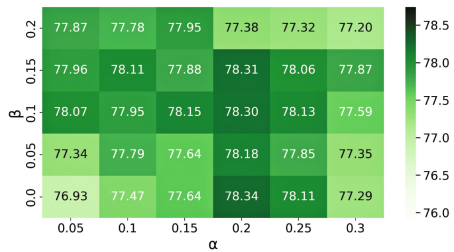
(a) Variation for similarity of anchor with various samples.



(b) Variation for similarity of inter-centroid and inter-cluster.



(c) Influence of hyperparameters Gaussian coefficient γ .



(d) Influence of hyperparameters of BML loss in CMNS-BERT_{base} ($\lambda=1e-4$).

Fig. 3. Multi perspective analysis of CMNS framework.

6.1 Sample Similarity

We depict the progression of similarity of the anchor x_i with the positive sample x_i^+ , false negatives C_{i1} , hard negatives C_{i2} , virtual negatives Z_i and in-batch

negatives during training in Figure 3a. Over the course of training, we observed significant fluctuations in the similarity between C_{i1} and x_i . It's obvious that the similarity of the anchor x_i with false negatives is much higher than other negative samples, highlighting the instability and uncertainty of false negatives. In contrast, the hard negatives we mined, as well as the virtual negatives generated based on them, demonstrate a stable trend of similarity during training and maintain a relatively high level of similarity. This stability is reliable and greatly beneficial for model training. On the other hand, in-batch negatives have extremely low similarity, making it difficult to provide valuable learning information to the model.

6.2 Cluster Trend

Furthermore, in Figure 3b, we plotted the evolution of average inter-centroid similarity and intra-cluster sample similarity. As training progresses, the similarity between cluster centroids gradually declines, indicating that clusters representing different semantics are slowly dispersing. This underscores the dynamic evolution of the clustering process and the refined segmentation of the semantic space. Simultaneously, the average intra-cluster similarity tends to stabilize, implying a higher degree of consistency in sentence embeddings among samples within each cluster. This evolution not only demonstrates the effectiveness of our clustering algorithm in capturing the inherent structure and characteristics of the data but also reflects the robustness and adaptability of the clustering process.

6.3 Noise-based Performance

In the CMNS framework, we add Gaussian noise to generate virtual negatives to diversify training data and boost model generalization. By adjusting the hyperparameter γ , we explore the effect of noise intensity on model performance. Figure 3c shows that too low noise (small γ) leads to redundancy and subpar performance. BERT achieves the best performance when γ is set in the range of [0.1-0.15], while RoBERTa and PromptBERT perform optimally when γ is set in the range of [0.25, 0.3]. Excessively high noise scatters data, reducing the value of virtual negatives for learning due to being too easy. Balancing the difficulty of negative samples is vital to optimizing model performance.

6.4 Hyperparameters in BML Loss

Moreover, we employ the BML loss function, meticulously tuning the hyperparameters α and β to regulate the semantic discrepancies among positive pairs, false negatives, and hard negatives. As depicted in Figure 3d, the selection of α is paramount, significantly impacting the discriminability between positive pairs and false negatives. Optimal model performance is achieved when α lies within [0.2, 0.25] and β falls between [0.1, 0.15]. This observation aligns with the similarity variation trend shown in Figure 3a, underscoring the importance of configuring semantic differences among sample pairs for our clustering method.

6.5 Transfer Task

Drawing inspiration from previous research, we have assessed our models using seven transfer tasks: MR, CR, SUBJ, MPQA, SST-2, TREC and MRPC [14, 19, 28, 29, 33, 34, 39]. For evaluation, we employed *Non-Prompt* CMNS models and adhered to the default settings provided by the SentEval Toolkit. The results presented in Table 3. Our model achieved higher performance compared to SimCSE in seven transfer tasks, and also showed significant improvement compared to other negative sample construction models.

Table 3. The accuracy results of various sentence embedding models for the transfer task are presented, with the best outcomes emphasized in bold.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg
GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
SimCSE-BERT _{base}	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
ClusterNS-BERT _{base}	80.98	85.78	94.53	88.95	85.94	88.20	74.55	85.56
CMNS-BERT _{base}	81.20	86.52	94.57	89.37	85.99	88.40	75.26	85.89
SimCSE-RoBERTa _{base}	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
DCLR-RoBERTa _{base}	82.47	86.86	93.48	87.96	87.14	84.80	74.16	85.27
ClusterNS-RoBERTa _{base}	81.78	86.65	93.21	87.85	87.53	84.00	76.46	85.35
CMNS-RoBERTa _{base}	81.44	87.76	93.17	87.37	87.54	86.80	75.64	85.67

6.6 Alignment and Uniformity

In order to thoroughly evaluate the sentence representation quality of our model CMNS, we adopt two metrics: Alignment and Uniformity [37]. Alignment measures the similarity between positive samples, indicating whether the model can map similar samples to close spatial positions:

$$\mathcal{L}_{align} \triangleq \mathbb{E}_{(x, x^+) \sim p_{pos}} \|f(x) - f(x^+)\|^2, \quad (7)$$

and uniformity assesses the uniformity of the representation space distribution, indicating whether the representations generated by the model are evenly distributed throughout the entire space:

$$\mathcal{L}_{uniform} \triangleq \log \mathbb{E}_{(x, y) \sim p_{data}} e^{-2\|f(x) - f(y)\|^2}. \quad (8)$$

Figure 4 illustrates the remarkable performance of the CMNS model in terms of alignment, preserving a high degree of similarity between positive samples and

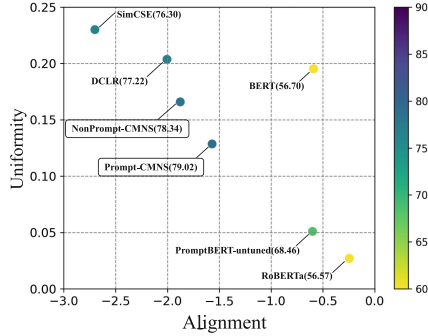


Fig. 4. Alignment and Uniformity of different sentence embedding models on the STS-B dataset. *untuned* means that the model has not undergone fine-tuning. All models above are based on BERT_{base}. The lower the value the better.

demonstrating commendable uniformity with representations distributed evenly across the space. In contrast, SimCSE lacks uniformity while RoBERTa overly focuses on uniformity, compromising sample similarity. This harmonious balance between maintaining sample similarity and ensuring representation space uniformity allows the CMNS model to exhibit robust performance across various tasks. Additional experiment analyses are provided in Appendix C.

7 Conclusion

In this paper, we introduce the CMNS framework, which aims to improve the quality of negative samples in unsupervised contrastive learning to optimize sentence representations. To achieve this goal, we integrate K-means clustering techniques into the training process and maintain a dynamically updated set of clusters using a Dynamic Cluster Queues Q . We select the second most similar cluster as hard negatives based on the similarity ranking between the anchor sentence and clusters in Q . Additionally, by introducing Gaussian noise to the centroid of the second most similar cluster, we can generate more high-quality negative samples. To reduce the impact of false negatives, we also introduce a bidirectional margin loss for constraint. Experimental results on STS tasks demonstrate significant improvements in performance. This work emphasizes the importance of enhancing negative sample quality in contrastive learning for sentence representations.

Acknowledgements. This work was supported in part by the National Key R&D Program of China (2023YFF1105102, 2023YFF1105105), the Major Project of the National Social Science Foundation of China (No. 21&ZD166), the National Natural Science Foundation of China (61876072) and the Natural Science Foundation of Jiangsu Province (No. BK20221535).

References

1. Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., 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)
2. Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J.: 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., Banea, C., Cer, D., Diab, M., Gonzalez Agirre, A., Mihalcea, R., Rigau Claramunt, G., Wiebe, J.: 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)
4. Agirre, E., Bos, J., Diab, M., Manandhar, S., Marton, Y., Yuret, D.: * 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). 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) (2012)
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. Bian, S., Zhao, W.X., Zhou, K., Cai, J., He, Y., Yin, C., Wen, J.R.: Contrastive curriculum learning for sequential user behavior modeling via data augmentation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 3737–3746 (2021)
7. Cai, T.T., Frankle, J., Schwab, D.J., Morcos, A.S.: Are all negatives created equal in contrastive instance discrimination? arXiv preprint [arXiv:2010.06682](https://arxiv.org/abs/2010.06682) (2020)
8. 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)
9. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
10. Conneau, A., Kiela, D.: Senteval: An evaluation toolkit for universal sentence representations. arXiv preprint [arXiv:1803.05449](https://arxiv.org/abs/1803.05449) (2018)
11. Deng, J., Wan, F., Yang, T., Quan, X., Wang, R.: Clustering-aware negative sampling for unsupervised sentence representation. arXiv preprint [arXiv:2305.09892](https://arxiv.org/abs/2305.09892) (2023)
12. Devlin, J.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint [arXiv:1810.04805](https://arxiv.org/abs/1810.04805) (2018)
13. Ding, N., Wang, X., Fu, Y., Xu, G., Wang, R., Xie, P., Shen, Y., Huang, F., Zheng, H.T., Zhang, R.: Prototypical representation learning for relation extraction. arXiv preprint [arXiv:2103.11647](https://arxiv.org/abs/2103.11647) (2021)

14. Dolan, B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Third international workshop on paraphrasing (IWP2005) (2005)
15. Du, B., Gao, X., Hu, W., Li, X.: Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3133–3142 (2021)
16. Ethayarajh, K.: How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. arXiv preprint [arXiv:1909.00512](https://arxiv.org/abs/1909.00512) (2019)
17. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint [arXiv:2104.08821](https://arxiv.org/abs/2104.08821) (2021)
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
19. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 168–177 (2004)
20. Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Towards unsupervised dense information retrieval with contrastive learning. arXiv preprint [arXiv:2112.09118](https://arxiv.org/abs/2112.09118) 2(3) (2021)
21. Jiang, T., Jiao, J., Huang, S., Zhang, Z., Wang, D., Zhuang, F., Wei, F., Huang, H., Deng, D., Zhang, Q.: Promptbert: Improving bert sentence embeddings with prompts. arXiv preprint [arXiv:2201.04337](https://arxiv.org/abs/2201.04337) (2022)
22. Kim, T., Yoo, K.M., Lee, S.g.: Self-guided contrastive learning for bert sentence representations. arXiv preprint [arXiv:2106.07345](https://arxiv.org/abs/2106.07345) (2021)
23. Li, J., Zhou, P., Xiong, C., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. arXiv preprint [arXiv:2005.04966](https://arxiv.org/abs/2005.04966) (2020)
24. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint [arXiv:1907.11692](https://arxiv.org/abs/1907.11692) (2019)
25. Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., Zamparelli, R.: Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014). pp. 1–8 (2014)
26. Miao, C., Cao, Z., Tam, Y.C.: Keyword-attentive deep semantic matching. arXiv preprint [arXiv:2003.11516](https://arxiv.org/abs/2003.11516) (2020)
27. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint [arXiv:1807.03748](https://arxiv.org/abs/1807.03748) (2018)
28. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. arXiv preprint [cs/0409058](https://arxiv.org/abs/cs/0409058) (2004)
29. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. arXiv preprint [cs/0506075](https://arxiv.org/abs/cs/0506075) (2005)
30. 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)
31. Robinson, J., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. arXiv preprint [arXiv:2010.04592](https://arxiv.org/abs/2010.04592) (2020)
32. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. *Advances in neural information processing systems* **30** (2017)

33. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642 (2013)
34. Voorhees, E.M., Tice, D.M.: Building a question answering test collection. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 200–207 (2000)
35. Wang, H., Dou, Y.: Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. In: International Conference on Intelligent Computing. pp. 419–431 (2023)
36. Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., Zhao, J.: Pico: Contrastive label disambiguation for partial label learning. In: International Conference on Learning Representations (2021)
37. 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)
38. Wang, W., Ge, L., Zhang, J., Yang, C.: Improving contrastive learning of sentence embeddings with case-augmented positives and retrieved negatives. arXiv preprint [arXiv:2206.02457](https://arxiv.org/abs/2206.02457) (2022)
39. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in. To appear in Language Resources and Evaluation **1**, 2 (2004)
40. Wu, X., Gao, C., Zang, L., Han, J., Wang, Z., Hu, S.: Esimcse: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. arXiv preprint [arXiv:2109.04380](https://arxiv.org/abs/2109.04380) (2021)
41. Zeng, J., Yin, Y., Jiang, Y., Wu, S., Cao, Y.: Contrastive learning with prompt-derived virtual semantic prototypes for unsupervised sentence embedding. arXiv preprint [arXiv:2211.03348](https://arxiv.org/abs/2211.03348) (2022)
42. Zhang, D., Nan, F., Wei, X., Li, S., Zhu, H., McKeown, K., Nallapati, R., Arnold, A., Xiang, B.: Supporting clustering with contrastive learning. arXiv preprint [arXiv:2103.12953](https://arxiv.org/abs/2103.12953) (2021)
43. Zhang, J., Lan, Z., He, J.: Contrastive learning of sentence embeddings from scratch. arXiv preprint [arXiv:2305.15077](https://arxiv.org/abs/2305.15077) (2023)
44. 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)

Author Index

A

Ahuja, Nilesh 48
Ambati, Shivanvitha 129
Anița, Ștefana-Lucia 407

B

Bellitto, Giovanni 331
Beyerer, Jürgen 249
Bhukya, Wilson Naik 129
Bilasco, Ioan Marius 299

C

Cao, Zongjing 1
Cheng, Zhen 423

D

Das, Debasis 392
De, Rajat K. 284
Ding, Ran 186
Dong, Zhiwei 186
Du, Bo 268

E

El-Assal, Mireille 299

F

Farahnakian, Fahimeh 32
Farahnakian, Farshad 32
Feillet, Eva 315
Fu, Hao 249
Fu, Penghao 144
Furnari, Antonino 65

G

Ghosh, Ayanabha 392
Gungor, Onat 48

H

Heikkonen, Jukka 32
Hemachandra, Nandyala 376

Hu, Yusong 156

Hua, Yang 440

Hudelot, Céline 315

J

Jain, Rishi 392

Jung, Cheolkon 144, 156, 171, 202

L

Li, Baoxin 97

Li, Ming 144, 156

Li, Mingyong 114

Li, Ruirui 16

Li, ShuPeng 362

Li, Wei 186

Li, Yan 1

Liu, Cheng-Lin 423

Liu, Yang 144, 156

Liu, Yunfeng 202

M

Maitra, Chayan 284

Manduca, Giovanni Maria 65

Maria Farinella, Giovanni 65

Mei, Lin 171

Mudgal, Priyanka 48

N

Ning, Weixun 234

O

Oh, Changjae 331

P

Padmanabhan, Vineet 129

Palaskar, Santosh 376

Palazzo, Simone 331

Pang, Yik Lung 331

Parida, Shubham 392

Paul, Riti 97

Pfrommer, Julius 249
 Popescu, Adrian 315
 Prasad Lal, Rajendra 129

Q

Qi, ZhaoHui 362

R

Rangaraj, Narayan 376
 Rios, Amanda 48
 Rosing, Tajana 48
 Rueda, Luis 81

S

Sajja, Surya Shraavan Kumar 376
 Sargolzaei, Saleh 81
 Sengupta, Anwasha 284
 Shah, Shivalee R. K. 347
 Sheikh, Javad 32
 Shin, Byeong-Seok 1
 Song, Xiaoning 440
 Sorrenti, Amelia 331
 Spampinato, Concetto 331
 Sun, Teng 234

T

Thakur, Nupur 97
 Tian, Long 331
 Tirilly, Pierre 299
 Turinici, Gabriel 218, 407

V

Vatsa, Mayank 347
 Vora, Sahil 97

W

Wang, Jiaxiang 186
 Wang, Kaige 249
 Wang, Naihao 16
 Wang, Yu 268
 Wang, Zun 440
 Wu, Chengzhi 249
 Wu, Shufan 268
 Wu, Xiao-jun 440

X

Xiao, Xiaoqiang 234

Y

Yang, Haixin 16
 Yang, Xingli 268
 Yang, YuKun 16

Z

Zelioli, Luca 32
 Zhang, Hao 171
 Zhang, Jiaming 249
 Zhang, Jie 114
 Zhang, Wenjie 440
 Zhang, Xu 234
 Zhang, Xu-Yao 423
 Zhang, Yuhang 440
 Zheng, Junwei 249
 Zhong, Zeyun 249
 Zhu, Fei 423