

Wenjie Zhang · Anthony Tung ·
Zhonglong Zheng · Zhengyi Yang ·
Xiaoyang Wang · Hongjie Guo (Eds.)

LNCS 14965

Web and Big Data

8th International Joint Conference, APWeb-WAIM 2024
Jinhua, China, August 30 – September 1, 2024
Proceedings, Part V

5 Part V



Springer

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.

Wenjie Zhang · Anthony Tung ·
Zhonglong Zheng · Zhengyi Yang ·
Xiaoyang Wang · Hongjie Guo
Editors

Web and Big Data

8th International Joint Conference, APWeb-WAIM 2024
Jinhua, China, August 30 – September 1, 2024
Proceedings, Part V

Editors

Wenjie Zhang 
University of New South Wales
Sydney, NSW, Australia

Zhonglong Zheng 
Zhejiang Normal University
Jinhua, China

Xiaoyang Wang 
University of New South Wales
Sydney, NSW, Australia

Anthony Tung 
National University of Singapore
Queenstown, Singapore

Zhengyi Yang 
University of New South Wales
Sydney, NSW, Australia

Hongjie Guo 
Zhejiang Normal University
Jinhua, China

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Computer Science

ISBN 978-981-97-7243-8

ISBN 978-981-97-7244-5 (eBook)

<https://doi.org/10.1007/978-981-97-7244-5>

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Singapore Pte Ltd. 2024

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 Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

If disposing of this product, please recycle the paper.

Preface

This volume (LNCS 14965) and its companion volumes (LNCS 14961, LNCS 14962, LNCS 14963, and LNCS 14964) contain the proceedings of the 8th Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, called APWeb-WAIM. Researchers and practitioners from around the world came together at this leading international forum to share innovative ideas, original research findings, case study results, and experienced insights in the areas of the World Wide Web and big data. The topics covered include web technologies, database systems, information management, software engineering, knowledge graphs, recommendation systems, and big data.

The 8th APWeb-WAIM conference was held in Jinhua from August 30 to September 1, 2024. As an Asia-Pacific flagship conference focusing on research, development, and applications related to Web information management, APWeb-WAIM builds on the successes of APWeb and WAIM. Previous APWeb conferences were held in Beijing (1998), Hong Kong (1999), Xi'an (2000), Changsha (2001), Xi'an (2003), Hangzhou (2004), Shanghai (2005), Harbin (2006), Huangshan (2007), Shenyang (2008), Suzhou (2009), Busan (2010), Beijing (2011), Kunming (2012), Sydney (2013), Changsha (2014), Guangzhou (2015), and Suzhou (2016). WAIM conferences were held in Shanghai (2000), Xi'an (2001), Beijing (2002), Chengdu (2003), Dalian (2004), Hangzhou (2005), Hong Kong (2006), Huangshan (2007), Zhangjiajie (2008), Suzhou (2009), Jiuzhaigou (2010), Wuhan (2011), Harbin (2012), Beidaihe (2013), Macau (2014), Qingdao (2015), and Nanchang (2016). The APWeb-WAIM conferences were held in Beijing (2017), Macau (2018), Chengdu (2019), Tianjin (2020), Guangzhou (2021), Nanjing (2022), and Wuhan (2023). With the ever-growing importance of appropriate methods in these data-rich times and the rapid development of web-related technologies, APWeb-WAIM will continue to be a flagship conference in this field.

The high-quality program documented in these proceedings would not have been possible without the authors who chose APWeb-WAIM for disseminating their findings. APWeb-WAIM 2024 received a total of 558 submissions. After the double-blind review process (each paper received at least three review reports), the conference accepted 149 regular research papers, 9 industry papers, and 13 demonstrations, resulting in an acceptance rate of 30.65%. The contributed papers address a wide range of topics, such as big data analytics, advanced database and web applications, data mining and applications, graph data and social networks, information extraction and retrieval, knowledge graphs, natural language processing, computer vision, generative AI and large language models, machine learning, recommender systems, security and blockchain, privacy and trust, and spatial and temporal data. We are grateful to these distinguished scientists for their invaluable contributions to the conference program.

We would like to express our gratitude to all individuals, institutions, and sponsors that supported APWeb-WAIM 2024. We are deeply thankful to the Program Committee members for lending their time and expertise to the conference. We also acknowledge

the support of the other members of the organizing committee, all of whom helped make APWeb-WAIM 2024 a success. We are grateful for the guidance of the honorary chair (Yunliang Jiang), the steering committee representative (Yanchun Zhang), and the general chairs (Qing Li, Kyuseok Shim, and Hong Gao) for their guidance and support. Thanks also go to the program committee chairs (Wenjie Zhang, Anthony K. H. Tung, and Zhonglong Zheng), industry chairs (Yaofeng Tu, Zhifeng Bao, and Wen Hua), demo chairs (Yajun Yang, Jing Jiang, and Chuan Xiao), workshop chairs (Yan Wang, Zhaoguo Wang, and Wenqi Fan), tutorial chairs (Xiang Zhao, Michael Sheng, and Xiangyu Zhao), publicity co-chairs (Bohan Li, Renata Borovica-Gajic, and Qian Zhou), publication chairs (Zhengyi Yang, Xiaoyang Wang, and Hongjie Guo), sponsorship chairs (Haofen Wang and Yong Liu), local chairs (Changjun Zhou and Lina Chen), CCF TCIS liaison (Xin Wang), and CCF TCDB liaison (Yueguo Chen).

August 2024

Wenjie Zhang
Anthony Tung
Zhonglong Zheng
Zhengyi Yang
Xiaoyang Wang
Hongjie Guo

Organization

Honorary Chair

Yunliang Jiang

Zhejiang Normal University, China

General Chairs

Qing Li

Hong Kong Polytechnic University, China

Kyuseok Shim

Seoul National University, South Korea

Hong Gao

Zhejiang Normal University, China

Program Committee Chairs

Wenjie Zhang

University of New South Wales, Australia

Anthony K. H. Tung

National University of Singapore, Singapore

Zhonglong Zheng

Zhejiang Normal University, China

Industry Track Chairs

Yaofeng Tu

Zhongxing Telecommunications Equipment,
China

Zhifeng Bao

Royal Melbourne Institute of Technology,
Australia

Wen Hua

Hong Kong Polytechnic University, China

Workshop Chairs

Yan Wang

Macquarie University, Australia

Zhaoguo Wang

Shanghai Jiao Tong University, China

Wenqi Fan

Hong Kong Polytechnic University, China

Tutorial Chairs

Xiang Zhao	National University of Defense Technology, China
Michael Sheng	Macquarie University, Australia
Xiangyu Zhao	City University of Hong Kong, China

Demo Chairs

Yajun Yang	Tianjin University, China
Jing Jiang	University of Technology Sydney, Australia
Chuan Xiao	Osaka University, Japan

Publicity Chairs

Bohan Li	Nanjing University of Aeronautics and Astronautics, China
Renata Borovica-Gajic	University of Melbourne, Australia
Qian Zhou	Nanjing University of Posts and Telecommunications, China

Publication Chairs

Zhengyi Yang	University of New South Wales, Australia
Xiaoyang Wang	University of New South Wales, Australia
Hongjie Guo	Zhejiang Normal University, China

Sponsorship Chairs

Haofen Wang	Tongji University, China
Yong Liu	Zhejiang Normal University, China

Local Chairs

Changjun Zhou	Zhejiang Normal University, China
Lina Chen	Zhejiang Normal University, China

CCF TCIS Liaison

Xin Wang

Tianjin University, China

CCF TCDB Liaison

Yueguo Chen

Renmin University of China, China

Steering Committee Representative

Yanchun Zhang

Zhejiang Normal University & Victoria
University, China & Australia

Program Committee Members

An Liu

Soochow University, China

Alexander Zhou

Hong Kong University of Science and
Technology, China

Bohan Li

Nanjing University of Aeronautics and
Astronautics, China

Bo Tang

Southern University of Science and Technology,
China

Baokang Zhao

National University of Defense Technology,
China

Bin Zhao

Nanjing Normal University, China

Chen Chen

University of Wollongong, Australia

Carson Leung

University of Manitoba, Canada

Conggai Li

CSIRO, Australia

Chenhao Ma

Chinese University of Hong Kong, China

Chuan Ma

Chongqing University, China

Cai Xu

Xidian University, China

Chuan Xiao

Osaka University, Nagoya University, Japan

Chengzhe Yuan

Guangdong Polytechnic Normal University, China

Chuanyu Zong

Shenyang Aerospace University, China

Deming Chu

University of New South Wales, Australia

Dianshu Liao

Australian National University, Australia

Dong Li

Liaoning University, China

Dongjing Miao

Harbin Institute of Technology, China

Dian Ouyang

Guangzhou University, China

Derong Shen	Northeastern University, China
Dejun Teng	Shandong University, China
Dong Wen	University of New South Wales, Australia
Dan Yin	Beijing University of Civil Engineering and Architecture, China
Donglin Zhu	Zhejiang Normal University, China
Faming Li	Northeastern University, China
Feiyi Tang	Guangzhou Panyu Polytechnic, China
Fan Zhang	Guangzhou University, China
Giovanna Guerrini	University of Genoa, Italy
Guanfeng Liu	Macquarie University, Australia
Guangxin Su	University of New South Wales, Australia
Guan Yuan	China University of Mining and Technology, China
Gengda Zhao	University of New South Wales, Australia
Harry Kai-Ho Chan	University of Sheffield, UK
Haipeng Dai	Nanjing University, China
Hong Gao	Nanjing University of Aeronautics and Astronautics, China
Hao Huang	Wuhan University, China
Huiqi Hu	East China Normal University, China
Hailong Liu	Northwestern Polytechnical University, China
Huan Li	Zhejiang University, China
Hui Li	Xidian University, China
Hiroaki Ohshima	University of Hyogo, Japan
Haiwei Pan	Harbin Engineering University, China
Hao Sun	University of Technology Sydney, Australia
Hanchen Wang	University of Technology Sydney, Australia
Hongzhi Wang	Harbin Institute of Technology, China
Haitao Yuan	Nanyang Technological University, Singapore
Jian Chen	Harbin Institute of Technology, China
Jun Gao	Peking University, China
Jiayi Liu	Harbin Institute of Technology, China
Junliang Li	Tianjin University, China
Jiali Mao	East China Normal University, China
Jianzhong Qi	University of Melbourne, Australia
Jie Shao	University of Electronic Science and Technology of China, China
Jiannan Wang	Simon Fraser University, Canada
Jianwei Wang	University of New South Wales, Australia
Jianzong Wang	Ping An Technology (Shenzhen) Co., Ltd., China
Jinbao Wang	Harbin Institute of Technology, China

Jun Wang	China University of Geosciences, China
Jiajie Xu	Soochow University, China
Jianing Xia	Deakin University, Australia
Jianqiu Xu	Nanjing University of Aeronautics and Astronautics, China
Jianke Yu	University of Technology Sydney, Australia
Jianye Yang	Guangzhou University, China
Jinguo You	Kunming University of Science and Technology, China
Junjie Yao	East China Normal University, China
Jia Zou	Arizona State University, USA
Jiujing Zhang	University of New South Wales, Australia
Junhua Zhang	University of New South Wales, Australia
Kaiyu Chen	University of New South Wales, Australia
Kongzhang Hao	Google, USA
Krishna Reddy P.	IIT, Hyderabad, India
Kai Wang	Shanghai Jiao Tong University, China
Kai Yao	Sichuan Normal University, China
Kaiqi Zhang	Harbin Institute of Technology, China
Luyi Bai	Northeastern University, China
Lizhen Cui	Shandong University, China
Lu Chen	Swinburne University of Technology, Australia
Lei Duan	Sichuan University, China
Li Jiajia	Shenyang Aerospace University, China
Lei Li	Hong Kong University of Science and Technology (Guangzhou), China
Longbin Lai	Alibaba Group, China
Liping Wang	East China Normal University, China
Long Yuan	Nanjing University of Science and Technology, China
Linhan Zhang	Oracle, USA
Muhammad Aamir Cheema	Monash University, Australia
Mizuho Iwaihara	Waseda University, Japan
Miaomiao Liu	Northeast Petroleum University, China
Mo Li	Liaoning University, China
Meng Wang	Southeast University, China
Michael Yu	Chinese University of Hong Kong, China
Ming Zhong	Wuhan University, China
Ning Liu	Shandong University, China
Nicolas Travers	Pôle Universitaire Léonard de Vinci, France
Peng Cheng	East China Normal University, China
Yaokai Feng	Kyushu University, Japan

Peiquan Jin	University of Science and Technology of China, China
Peng Wang	Fudan University, China
Peilun Yang	Zhejiang Lab, China
Qiuyu Guo	University of New South Wales, Australia
Qi Luo	University of New South Wales, Australia
Qingqiang Sun	Great Bay University, China
Qiuyan Yan	China University of Mining and Technology, China
Roshni Iyer	University of California, Los Angeles, USA
Ruihong Qiu	University of Queensland, Australia
Rui Zhu	Shenyang Aerospace University, China
Sara Comai	Politecnico di Milano, Italy
Shunyang Li	University of New South Wales, Australia
Sanjay Madria	Missouri University of Science & Technology, USA
Sanghyun Park	Yonsei University, South Korea
Shidong Pan	Australian National University & CSIRO's Data61, Australia
ShiJie Sun	Chang'an University, China
Shuai Xu	Nanjing University of Aeronautics and Astronautics, China
Shanshan Yao	Shanxi University, China
Shiyu Yang	Guangzhou University, China
Shuigeng Zhou	Fudan University, China
Shuiqiao Yang	CSIRO, Australia
Taotao Cai	University of Southern Queensland, China
Tung Kieu	Aalborg University, Denmark
Tianming Zhang	Zhejiang University of Technology, China
Wang Lizhen	Yunnan University, China
Wei Li	Harbin Engineering University, China
Wenpeng Lu	Qilu University of Technology, China
Wentao Li	Hong Kong University of Science and Technology (Guangzhou), China
Wei Shen	Nankai University, China
Wei Song	Wuhan University, China
Weiguo Zheng	Fudan University, China
Wen Zhang	Wuhan University, China
Xin Bi	Northeastern University, China
Xin Cao	University of New South Wales, Australia
Xuefeng Chen	Chongqing University, China
Xiaofeng Ding	Huazhong University of Science and Technology, China

Xiaou Ding	Harbin Institute of Technology, China
Xiaolin Fang	Southeast University, China
Xinwei Jiang	China University of Geosciences, China
Xiang Lian	Kent State University, USA
Xueli Liu	Tianjin University, China
Xiangfu Meng	Liaoning Technical University, China
Xiao Pan	Shijiazhuang Tiedao University, China
Xuguang Ren	MBZUAI, UAE
Xiangyu Song	Swinburne University of Technology, Australia
Xiaohui (Daniel) Tao	University of Southern Queensland, Australia
Xingyu Tan	University of New South Wales, Australia
Xiaoyang Wang	University of New South Wales, Australia
Xin Wang	Tianjin University, China
Xubo Wang	University of Technology Sydney, Australia
Xiaojun Xie	Nanjing Agricultural University, China
Xiaochun Yang	Northeastern University, China
Xiang Zhao	National University of Defense Technology, China
Xiangmin Zhou	RMIT University, Australia
Xiao Zhang	Shandong University, China
Xiaowang Zhang	Tianjin University, China
Xuliang Zhu	Shanghai Jiao Tong University, China
Xuyun Zhang	Macquarie University, Australia
Xujian Zhao	Southwest University of Science and Technology, China
Yunpeng Chai	Renmin University of China, China
Yixiang Fang	Chinese University of Hong Kong, Shenzhen, China
Yanhui Gu	Nanjing Normal University, China
Yunjun Gao	Zhejiang University, China
Yiheng Hu	University of New South Wales, Australia
Yihong Huang	East China Normal University, China
Yi Jin	Data Principles (Beijing) Technology Co., Ltd., China
Yongchao Liu	Ant Group, China
Yu Liu	Huazhong University of Science and Technology, China
Yu Liu	Beijing Jiaotong University, China
Yang-Sae Moon	Kangwon National University, South Korea
Yuwei Peng	Wuhan University, China
Yu-Xuan Qiu	Beijing Institute of Technology, China
Yongpan Sheng	Southwest University, China

Yifu Tang	Deakin University, Australia
Yiping Teng	Shenyang Aerospace University, China
Yong Tang	South China Normal University, China
Yanping Wu	University of Technology Sydney, Australia
Yaoshu Wang	Shenzhen Institute of Computing Sciences, Shenzhen University, China
Yikun Wang	University of New South Wales, Australia
Yuanbo Xu	Jilin University, China
Yuanyuan Xu	University of New South Wales, Australia
Yajun Yang	Tianjin University, China
Yuanhang Yu	University of Technology Sydney, Australia
Yanfeng Zhang	Northeastern University, China
Yong Zhang	Tsinghua University, China
Yongqing Zhang	Chengdu University of Information Technology, China
Youwen Zhu	Nanjing University of Aeronautics and Astronautics, China
Yuanyuan Zhu	Wuhan University, China
Yuxiang Zeng	Beihang University, China
Zouhaier Brahmia	University of Sfax, Tunisia
Zemin Chao	Harbin Institute of Technology, China
Zi Chen	Nanjing University of Aeronautics and Astronautics, China
Zihan Feng	Tianjin University, China
Ziquan Fang	Zhejiang University, China
Zhao Li	Tianjin University, China
Zhen Tao	Australian National University, Australia
Zhaokang Wang	Nanjing University of Aeronautics and Astronautics, China
Zhibin Wang	Nanjing University, China
Zhuoran Wang	University of New South Wales, Australia
Zhengyi Yang	University of New South Wales, Australia
Zihan Yang	University of Melbourne, Australia
Ziqiang Yu	Yantai University, China
Zhaonian Zou	Harbin Institute of Technology, China
Zixu Zhao	University of New South Wales, Australia

Contents – Part V

Anomaly Detection and Security

TWLog: Task Workflow-Based Log Anomaly Detection	3
<i>Suqiong Zhang, Dongyi Fan, Lili He, Yi Liu, and Deng Chen</i>	
TS-AUBD: A Novel Two-Stage Method for Abnormal User Behavior Detection	17
<i>Yu Cao, Yilu Chen, Ye Wang, Ning Hu, Zhaoquan Gu, and Yan Jia</i>	

Multi-label Out-of-Distribution Detection with Spectral Normalized Joint Energy	31
<i>Yihan Mei, Xinyu Wang, Dell Zhang, and Xiaoling Wang</i>	

Noisy Label Learning Based on Weighted Neighborhood Consistency	46
<i>Qian Rong, Lu Zhang, Ling Yuan, Xuanang Ding, and Guohui Li</i>	

Information Retrieval

A New Learning-to-Rank Framework for Keyphrase Extraction Using Multi-scale Ratings and Feature Fusion	63
<i>Corina Florescu, Avijeet Shil, and Wei Jin</i>	

MIIGraph: Multi-granularity Information Integration Graph for Document-Level Event Extraction	80
<i>Lin Mu, Yide Cheng, Xiaoyu Wang, Yang Li, and Yiwen Zhang</i>	

Multi-granularity Neural Networks for Document-Level Relation Extraction	95
<i>Xiye Chen and Peng Wang</i>	

Improving Zero-Shot Information Retrieval with Mutual Validation of Generative and Pseudo-Relevance Feedback	113
<i>Xinran Xie, Rui Chen, TaiLai Peng, Dekun Lin, and Zhe Cui</i>	

Entity Semantic Feature Fusion Network for Remote Sensing Image-Text Retrieval	130
<i>Jianan Shui, Shuaipeng Ding, Mingyong Li, and Yan Ma</i>	

Semantic Preservation and Hash Fusion Network for Unsupervised Cross-Modal Retrieval	146
<i>Xinsheng Shu and Mingyong Li</i>	
Machine Learning	
Using High-Quality Feature for Weakly-Supervised Camouflaged Object Detection	165
<i>Weijie Wu, Yiqiu Tong, Qijun Jiang, Lina Chen, and Hong Gao</i>	
ECHO: Adaptive Correction for Subgraph-Wise Sampling with Lightweight Hyperparameter Search	179
<i>Dingwei Liu, Zhenyu Li, and Zhibin Zhang</i>	
A Parallel and Distributed Data Management Approach for MEC Using the Improved Parameterized Deep Q-Network	195
<i>Bingqing Ren, Peng Yang, Meng Yi, and Dongmei Yang</i>	
Clustering Based Collaborative Learning Grouping for Knowledge Building	210
<i>Jiaqi Hao, Weipo Yi, Meirui Ren, Chunyu Ai, Tianlong Qi, and Longjiang Guo</i>	
Unsupervised Feature Selection via Fuzzy K-Means and Sparse Projection	224
<i>Kun Jiang, Lei Zhu, and Qindong Sun</i>	
Open World Semi-supervised Learning Based on Multi-scale Enhanced Feature	240
<i>Tianming Zhang, Kejia Zhang, Haiwei Pan, and Yuechun Feng</i>	
ACD: Attention Driven Cognitive Diagnosis for New Learners Joining ITS	255
<i>Bingdi Shao, Keai Wei, Longjiang Guo, Meirui Ren, Lichen Zhang, and Peng Li</i>	
Data Augmentation for Knowledge Tracing Based on Variational AutoEncoder and Efficient Network Reusing	271
<i>Hui Zhao and Jun Sun</i>	
An Epidemic Trend Prediction Model with Multi-source Auxiliary Data	286
<i>Benfeng Wang, Xiaohua He, Hang Lin, Guojiang Shen, and Xiangjie Kong</i>	
Lead-Aware Hierarchical Transformer and Convolution Fusion Network for ECG Classification	302
<i>Yuang Zhang, Binyu Wang, Liping Wang, and He Huang</i>	

Reinforcement Learning from Clip	318
<i>Shaoqiang Zhu, Kejia Zhang, and Haiwei Pan</i>	
Self Supervised Contrastive Learning Combining Equivariance and Invariance	330
<i>Longze Yang, Yan Yang, and Hu Jin</i>	
Demonstration Paper	
FedPPQs: Optimizing Property Path Queries Evaluation over Federated RDF Systems	347
<i>Jibing Wu, Ningchao Ge, Tengyun Wang, Xuan Li, Lihua Liu, and Hongbin Huang</i>	
MPCPM: Multi-level Prevalent Co-location Pattern Miner	352
<i>Vanluan Nguyen and Vanha Tran</i>	
FGAQ: Accelerating Graph Analytical Queries Using FPGA	357
<i>Yi Ding, Zhengyi Yang, Shunyang Li, Liuyi Chen, Haoran Ning, Kongzhang Hao, and Yongfei Liu</i>	
A Progressive Question Answering Framework Adaptable to Multiple Knowledge Sources	362
<i>Yirui Zhan, Yanzeng Li, Minhao Zhang, and Lei Zou</i>	
RocolSys: An Automatic Row-Column Data Storage System for HTAP	368
<i>Shuangshuang Cui, Hongzhi Wang, Hao Wu, Dong Wang, Jinxuan Li, Jingbiao Ren, Chenguang Li, and Wei Zhao</i>	
MIPC-SHOPs: An Online System for Mining the Influence of Industrial Pollution on Cancer Based on the Spatial High-Influence Ordered-Pair Patterns	373
<i>Lingli Zhang, Lizhen Wang, Peizhong Yang, and Lihua Zhou</i>	
A Perception System for DNS Root Service Status Based on Active and Passive Monitoring	378
<i>Guozhong Dong, Hao Guo, and Hualong Wu</i>	
Dynamic Route Planning System Integrated with Traffic Flow Sensing	383
<i>Bingkun Wang, Yixin Tian, Fangshu Chen, Jiahui Wang, and Yufei Zhang</i>	
NLITS: A Natural Language Interface for Time Series Databases	388
<i>Yuting Lin, Jianqiu Xu, Xieyang Wang, and Yitong Zhang</i>	

FOICP-Miner: An Interactive Spatial Pattern Recommendation System Based on Fuzzy-Ontology	393
<i>Zhiwei Chen, Zezheng Geng, and Xuguang Bao</i>	
SPCCP-Miner: Towards the Discovery of Congested Junctions	398
<i>Zheyang Liu, Zhengyu Yang, and Xuguang Bao</i>	
Crowd-OBIGA: A Crowdsourced Approach for Oracle Bone Inscriptions Glyph Annotation	403
<i>Zhaohan Dong, Xiaofan Wang, Jing Xiong, Guangshun Li, and Qingju Jiao</i>	
NexusDB: A Large-Scale Distributed Time-Series Database for Industrial Scenarios	408
<i>Linlin Ding, Di Yuan Chzhen, Yuda Li, Zhiyong Zhang, Zhiran Xie, and Mo Li</i>	
Industry Paper	
LMStor: Storage Acceleration Design for Large Models	415
<i>Biyun Shang, Feng Zhang, Mo Xu, Junning Xu, and Zhenjiang Dong</i>	
Enhancing Emergency Communications via UAV-Assisted Home-Independent Broadband Mobile Networks	427
<i>Yiping Zhang and Haobin Shi</i>	
FPTSF: A Failure Prediction of Hard Disks Based on Time Series Features Towards Low Quality Dataset	438
<i>Xiaoyu Lu, Chenfeng Tu, Hongzhang Yang, Jiangpu Guo, and Hailong Sun</i>	
PMEMgreSQL: Embracing PostgreSQL with Persistent Memory	448
<i>Xinyuan Sun, Ji Shi, Yinjun Han, and Zhenghua Chen</i>	
The Development of a TLA ⁺ Verified Correctness Raft Consensus Protocol ...	459
<i>Hua Guo, Yunhong Ji, and Xuan Zhou</i>	
Robust Multi-vehicle Routing with Communication Enhanced Multi-agent Reinforcement Learning for Last-Mile Logistics	470
<i>Hai Wang, Shuai Wang, Shuai Wang, and Xiaolei Zhou</i>	
A Dual-Tower Model for Station-Level Electric Vehicle Charging Demand Prediction	481
<i>Qinyuan Li, Lei Yao, Shaolin Wang, Haoyang Che, and Yan Yi</i>	

BPGNNSBR: Behavior Progressive Graph Neural Networks for Session-Based Recommendation	492
<i>Zekun Xu, Wenlong Wu, Zhanzuo Yin, Xinzhe Zhao, Junnan Zhuo, and Bohan Li</i>	
Exploring Simple Architecture of Just-in-Time Compilation in Databases	504
<i>Haoran Ning, Bocheng Han, Zhengyi Yang, Kongzhang Hao, Miao Ma, Chunling Wang, Boge Liu, Xiaoshuang Chen, Yu Hao, Yi Jin, Wanchuan Zhang, and Chengwei Zhang</i>	
Author Index	515

Anomaly Detection and Security



TWLog: Task Workflow-Based Log Anomaly Detection

Suqiong Zhang^(✉) , Dongyi Fan, Lili He, Yi Liu, and Deng Chen

School of Computer Science and Technology, Zhejiang Sci-Tech University,
Hangzhou 310018, China

`zhangsuqiong22@gmail.com`, `{2023110602001, 202210602002,`
`202210602001}@mails.zstu.edu.cn`, `llhe@zju.edu.cn`

Abstract. Log anomaly detection is crucial for pinpointing software issues, particularly in large-scale distributed systems where numerous services span multiple machines. Logs are abundant in such environment, meanwhile the log sequences lack inherent business logic when examined solely based on timestamps. Moreover, the parallel execution of tasks leads to interleaved logs which caused log entries belonging to same task span a wide range of timestamps. Existing anomaly detection approaches ignore the complex structure of a trace brought by its invocation hierarchy. In addition, most of them based on RNNs which is difficulty in capturing long-range dependencies and the challenge of handling variable-length sequences effectively. In this paper, we propose TWLog, a self-supervised deep learning-based method. TWLog uses a unified trace graph to describe the complex structure of a trace combining the task workflow and log events. Based on the basic task workflow from log message, we extract the semantic information from raw log messages as vector representations. These vectors are then fed into a Transformer-based model which can capture the contextual information from task workflow-based log sequences. The experiments on real-world dataset of HDFS, OpenStack and Kubernetes confirm the effectiveness of our method.

Keywords: Microservices · Log Analysis · Deep Learning

1 Introduction

Large distributed systems generate interleaved logs from multiple components and services, making anomaly detection challenging due to the complexity and volume of log data. These logs often contain diverse information related to system events, errors, and activities across various layers of the distributed infrastructure. Log anomaly detection plays an important role nowadays. It can distinguish the normal and abnormal behavior, identify patterns indicative of anomalies.

Currently, numerous traditional machine learning models have been developed to detect anomalous events from log messages. These approaches typically

involve extracting relevant features from the log messages and applying conventional machine learning algorithms for log data analysis. These traditional models lack the capability to understand the sequential order and timing of log events, which is crucial for detecting anomalies in log data. As a result, deep learning models, especially recurrent neural networks (RNNs), are widely used for log anomaly detection since they can capture the temporal information in sequential data [1, 7, 8].

However, there are still some limitations of using RNN for modeling log data. One major limitation is the difficulty in capturing long-term dependencies and context information in log sequences, especially when dealing with logs from distributed systems where events may be widely separated in time [1]. It is crucial to observe the complete context information instead of only the information from previous steps when detecting malicious attacks based on log messages. Additionally, RNNs may struggle with handling variable-length sequences and may require padding or truncation strategies, which could introduce biases or information loss in the modeling process. These challenges highlight the need for more advanced approaches that can overcome the limitations of RNNs and better leverage the rich temporal information present in log data.

Another critical problem is that service or microservice introduce unique challenges for anomaly detection especially in large-scale distributed systems. The structure of traces within such systems can be intricate, shaped by the hierarchy of service invocations. A trace can include several to hundreds of service invocations. Industrial distributed tracing systems can link the log messages of the same trace by injecting the identify the log messages produced by different service instances [26]. The inherent complexity of these services leads the uncertainty of the log sequence for one task workflow from timestamp. Traditional anomaly detection methods often missed the business flow's invocation relationships with fixed windows size. Moreover, for distributed system, log sequence pattern ordered by timestamp is unstable due to the parallel execution among instances and requests. Log entry is produced for each request from different nodes involved in the request by timestamp.

To overcome above limitations of existing approaches, we propose TWLog in this paper, a self-supervised method for log anomaly detection. Inspired by the great success of Transformer in modeling sequential text data [1], we leverage Transformer-based model to capture patterns of normal log sequences. The Transformer-based model with multi-head self-attention mechanism can learn the contextual information from the log sequences with various lengths in the form of vector representations. Furthermore, we use a unified graph representation, which is called trace relation graph, to describe the complex structure of a trace combining the task workflow and log events. Then, log entries based on trace relation is generated to the vector representations and fed into the anomaly detection model. When an anomaly is detected, TWLog raises an alarm to users. Note that TWLog does not depend on the log labelling, it just requires the normal log sequence extracted from the system. We have evaluated the proposed approach on one public and local dataset. The experimental results show that our

method can achieve stable performance in real world log dataset. The overview of the proposed approach is shown in Fig. 1.

In summary, this paper makes the following contributions:

- Based on task workflow, we construct the trace relation graphs. The log sequence combine with the trace relation graph can enhance the representation of log data for anomaly detection.
- Employing Transformer-based models for anomaly detection, leveraging their ability to capture long-range dependencies and global context in log sequences effectively.

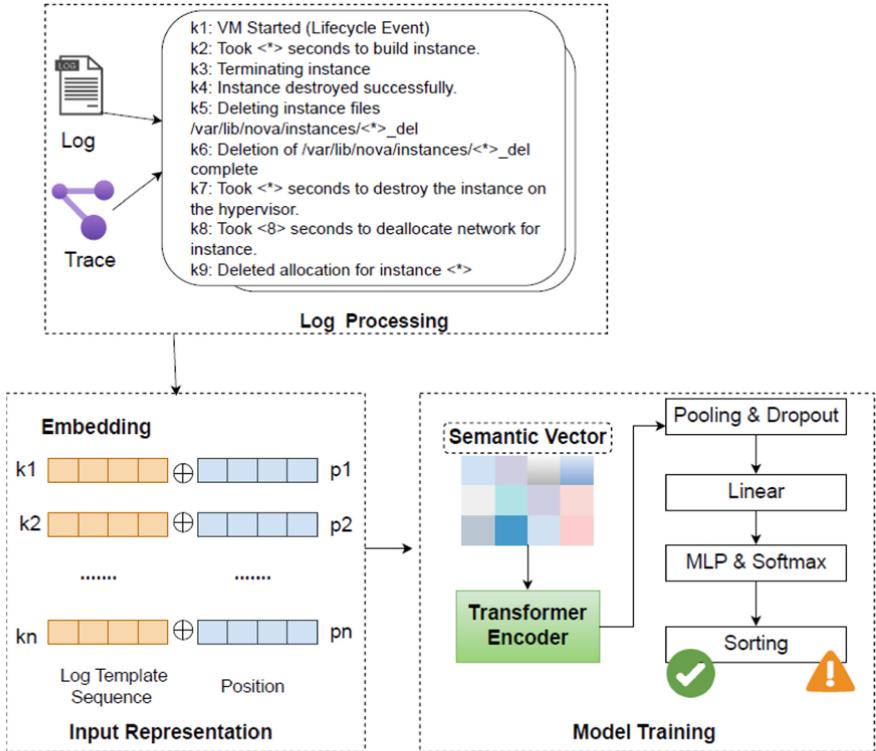


Fig. 1. Overview of TWLog.

2 Background and Motivation

2.1 Background

Large and intricate software-intensive systems frequently generate substantial volumes of log data to facilitate troubleshooting during system operation. Log

data captures the events and internal states of the system during runtime. Through log analysis, operators gain insights into the system's status and can effectively diagnose issues when failures occur. Each log entry typically includes contextual information such as the component name, timestamp, severity level, and unique request or transaction ID. This helps correlate logs across different services and trace requests through the system. In large-scale distributed systems, distributed tracing [2] is widely used to profile and monitor their executions. A trace is the description of the execution process of a request as it moves through a distributed system. These tracing identifiers can be used to group log entries together or untangle log entries produced by concurrent processes to separate, single-thread sequential sequences [3–5]. Each ID identifies an abstracted concept or a concrete resource instance. Certain relationships reflected in the corresponding log messages. With these IDs, we can reconstruct the interaction between components from log. Unique values may serve as ID for request or task execution, e.g., blockid in HDFS log and instanceid in OpenStack log.

In distribute service system, one business task is a dedicated workflow that we can extract from log message. For instance, the lifecycle of OpenStack VM (Virtual Machine), the instance starts from VM creation, VM stop to VM deletion and other query requests. These tasks have clear and strict sequential order. VM stop triggered after VM creation finished. But log message with same key message appears in different tasks. E.g. “VM Resumed (Lifecycle Event)” may appear during different tasks among VM creation, VM start, VM resume. It leads the uncertainty of the log sequence for one task. Figure 2 shows one simple task workflow during VM stop.

1: Terminating instance
2: Instance destroyed successfully.
3: Deleting instance files /var/lib/nova/instances/<*>_del
4: Deletion of /var/lib/nova/instances/<*>_del complete
5: Took <*> seconds to destroy the instance on the hypervisor.
6: Took <*> seconds to deallocate network for instance.
7: Deleted allocation for instance <*>
8: VM Stopped (Lifecycle Event)

Fig. 2. OpenStack VM stop workflow example.

2.2 Motivation

In modern software applications, the task workflow can be recorded by developers to analysis the code from business view. Generally, the relationship between components is complicated due to the concurrently running threads inside each task. Figure 3 shows the timeline of requests invocation of instances in OpenStack log messages. There are two instances with ID A and ID B in this example. The lifecycle management for both of them is from instances creation to instances

termination. Instance A is firstly created, then it invokes request A1 for system resource query such as vcpus, network query. Request A2 is sent after request A1 finished. Meanwhile, users trigger request to create instance B, due to the parallel request, some overlap between instance A and B exists. In other words, when instance B creates, instance A still alive. Similar parallel execution also exists among requests. From Fig. 3, request B2 invokes before request B1 finish which cause the overlap in the timeline for requests. In short word, the log sequence pattern ordered by timestamp is unstable due to the parallel execution among instances and requests.

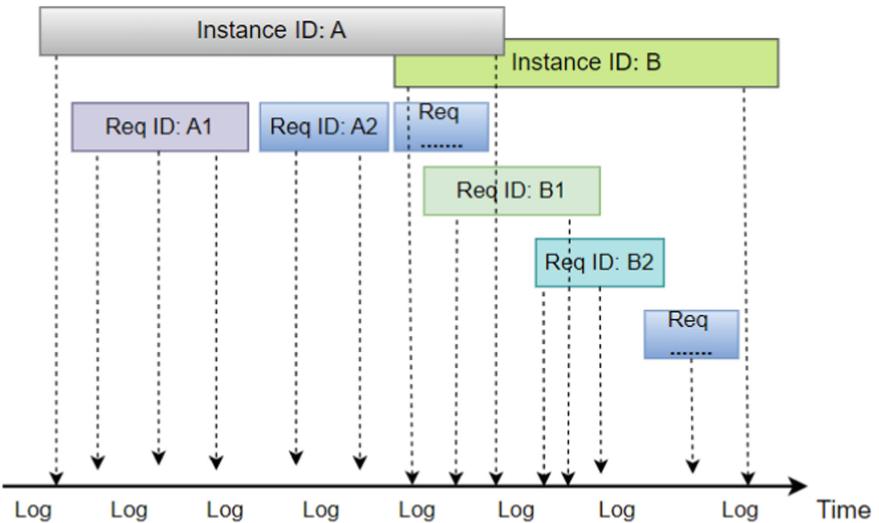


Fig. 3. Timeline of requests invocation in OpenStack log.

The existing approaches primarily focus on detecting anomalies within log sequences by correlating them with a single component [25] or solely with a timeline. To effectively localize anomalies for distributed systems, it is crucial to explore dependencies between unique identifiers (IDs) and correlate log messages associated with the same ID across distributed components. Therefore, our objective is to extract normal log sequence patterns for specific tasks and construct a workflow model for each task based on its log key sequence. This comprehensive approach enables a more thorough understanding of the distributed system's behavior and facilitates the accurate detection of anomalies across multiple components.

3 Approach

3.1 Relation Graph Construction

Logs are unstructured text generated according to the definition of log statements in source code. We firstly study the characteristics of general log format in different systems and software [6], we found that there is common format for log messages. The format can be expressed as: “<Timestamp> <Log level> <Owner> [Sub owner] <Log level> <Event ID> [Sub event ID] <Log content>”.

In this paper, we first extract log features from information, then build the log sequence pattern for dedicated task according to the task trace invocation. We finally regroup the log entries based on the relation graph and the timestamp of log events. Take OpenStack nova log as an example, firstly we construct the basic task workflow model. G is a directed acyclic graph (DAG), where each node represents the owner and ID, each edge captures the hierarchical relationship among owners and IDs. The relation can be 1:n, m:n. The empty relation stands for that there is no dependency between two owners (IDs). 1:n means the object dispatches n objects. For instance, there are sub owners under owner nova, such as nova.compute.manager which is responsible for executing tasks by interacting with other nova components. The relation m:n indicates that m request ID are used for handling n different tasks. Figure 4 is the abstraction of the task workflow. We obtain the basic service invocation relationships through log analysis or prior knowledge.

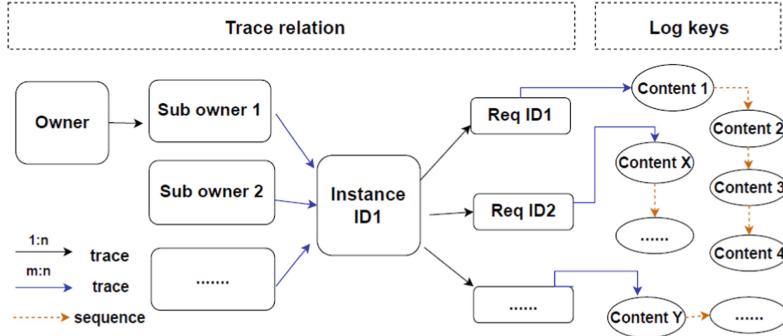


Fig. 4. Relation graph based on abstraction of task workflow.

3.2 Log Embedding

Log Processing. We first extract log keys (templates) from log entries. Log keys represent the structured components of log messages for understanding the structure of log data. Additionally, due to log message contain non-character tokens such as separators like “/”, dashes, and various symbols. We need to process log

entries by removing non-verbal symbols and stop word according to previous works [10, 11] and splitting compound words into individual words. For example, the log key “Attempting claim on node wally123.cit.tu-berlin.de: memory 4096 MB, disk 40 GB, vcpus 2 CPU” is processed to “Attempting claim on node <*> memory <*> disk <*> vcpus <*>”.

Word Embedding. First, each log key is split into a set of tokens by the whitespace character and build a vocabulary from these tokens. We adopt the WordPiece tokenization [29], which is widely used in many recent language modeling studies [13]. It trains language model starting from the base vocabulary and picks the pair with the highest likelihood. This pair is added to the vocabulary, and the language model is again trained on the new vocabulary. These steps are repeated until the desired vocabulary is reached [14]. Words of log key are further transformed into vectors by using Word2Vec [30].

Sentence Embedding. Sentence embedding generates a vector representation for each log event based on word embedding. The role of words in log key is not equally important. Some words are more important than others in sentence embedding. Thus, we use TF-IDF to give the weight of each word in sentence [16]. Word for TF (Term Frequency) w in a log key k measures the importance in log message. It can be calculated by $TF_{w,k} = \frac{N_{W,k}}{N_k}$, here $N_{W,k}$ is the number of word w in log key k and N_k is the total number of words in log key k . While IDF (Inverse Document Frequency) of a word w measures the frequency in all log keys. IDF calculates by $IDF_w = \log \frac{N}{N_w}$, N is the total log keys and N_w is the number of log key which contains w . The weight of word w in log key k can be measured by $TF - IDF$ score: $S_{w,k} = TF_{w,k} \times IDF_w$. The log key vector representation is the weighted sum of vector of all its words as following:

$$V_e = \frac{1}{C_k} \sum_{w=1}^{c_k} W_{w,k} \cdot V_w \quad (1)$$

Here C_k is the total count of unique words in k and V_w is the vector representation of word w . Due to the word characteristics in log key and trace relation, we calculate TF-IDF and weight for log key and trace relation separately.

3.3 Neural Representation

After task workflow construction, we have the relation graph. Based on this relationship, we get the task log sequence which is ordered by log keys. We define log sequence as $S = \{l_1, l_2, \dots, l_t, \dots, l_n\}$, where $l_t \in L$ indicates the log key in t^{th} position, L is a set of log keys processed from log message. Our goal is to predict whether new log sequence L is anomalous according to the train dataset. The train dataset defined as $D = \{L^j\}_{j=1}^N$ only consists of normal log entries. For a normal log sequence L^j , we add a special token CLS in the beginning of L , which is used to represent the whole log sequence based on the structure of Transformer encoder. We will use the contextual embedding of CLS to constraint the

distribution of normal log sequences. We define $S^j = \{l_1^j, \dots, l_t^j, \dots, l_n^j\}$ is the j^{th} log sequence. Each log key l_t is the input representation of x_t^j , x_t^j is the sum of log key embedding and position embedding. Log key embedding is generated in section of event embedding. In this paper, matrix $E \in \mathbb{R}^{|K| \times d}$ is the log key embedding matrix where d is the dimension of log key embedding.

The contextual information for log sequential data is captured by injection the position information of log keys in the log sequence. We denote the position embedding as $P \in \mathbb{R}^{P \times d}$ which has the same dimension with log key embedding. In addition, we use the same sinusoid function as the position encoding to generate position embedding [9]. It is defined as $T_{t,2i} = \sin\left(\frac{t}{100002i/d}\right)$; $T_{t,2i+1} = \cos\left(\frac{t}{100002i/d}\right)$, t is the t^{th} position in the log sequence; i is the i^{th} dimension in the d dimension embedding. For any fixed offset k , P_{t+k} can be represented as a linear function of P_t which is easily to learn to attend by relative positions. Lastly, the neural representation can be defined as:

$$X_t^j = E_{L_t^j} + P_{L_t^j} \quad (2)$$

3.4 Transformer Encoder

Transformer architecture [9] contains self-attention layers followed by position-wise feed-forward layers which can get the contextual relations among log keys in log sequence. Each transformer layer includes a multi-head self-attention sub-layer and a position-wise feed forward sub-layer. Multi-head attention layers calculate the attention score for each log entries with different attention patterns. The attention score is calculated by training the query and key matrices of the attention layers. Formally, for the i^{th} head of the attention layer, the scaled dot-product self-attention is defined as:

$$\text{head}_i = \text{Attention}(X^j W_i^Q, X^j W_i^K, X^j W_i^V) \quad (3)$$

where $\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_v}}\right)V$; $X_j \in \mathbb{R}^{T \times d}$ is the input representation of the log sequence; W_i^Q, W_i^K, W_i^V is the linear projection weights. Each self-attention operation enables each key to attend to all log keys within an input sequence, calculating the hidden representation for each log key based on an attention distribution over the entire sequence. The position-wise feed-forward sub-layer, which incorporates a ReLU activation function, is independently applied to the hidden representation of each activity. Moreover, by integrating both the position-wise feed-forward sub-layer and multi-head attention mechanisms, a transformer layer is defined as:

$$\text{transformer_layer}(X^j) = \text{FFN}(f(X^j)) = \text{ReLU}(f(X^j)W_1)W_2 \quad (4)$$

where W_1 and W_2 are trained projection matrices. The Transformer encoder usually consists of multiple transformer layers. We use h_t^j as the contextual embedding vector for log keys which is produced by the Transformer encoder.

$$h_t^j = \text{transformer}(x_t^j) \quad (5)$$

3.5 Anomaly Detection

When a new set of log arrives, firstly it conducts the log processing to get log sequence with trace relation. Then the task workflow based log sequence keys are generated. By log embedding, log keys is transformed to semantic vectors which fed into the trained model. Furthermore, the transformer based model predict whether there is anomalous in the log sequence. The normal or abnormal log sequence is identified by softmax classifier. Similar to DeepLog [25], a candidate log set is built which consists of log keys with most m probable log entries. The probabilities are the outputs of the softmax function. For example, we want to confirm if the log entries l_t is normal, a window size log keys $M_h = \{m_{t-h}, m_{t-h+1}, \dots, m_{t-1}\}$ is sent to the trained model. The corresponding output are the probability distributions:

$$P(l_t|W) = \{l_1 : p_{l1}, l_2 : p_{l2}, \dots, l_n : p_{ln}\} \quad (6)$$

If the real log l_t is not in the probabilities $P(l_t|W)$, then it will be treated as anomaly. An alarm will be raised to users.

4 Evaluation

The experiments in this paper are executed on a Linux server with 20 cores CPU, 128 GB RAM, and one NVIDIA Tesla P4 GPU. The software version we used is Python 3.9.18 and PyTorch 1.9.0. To evaluate our solution (TWLog), we conduct experimental studies to answer the following research questions:

- **RQ1:** How effective is TWLog in log anomaly detection compared with baseline approaches?
- **RQ2:** How much does the trace relation graph contribute to the effectiveness in TWLog?
- **RQ3:** How do different configurations impact the effectiveness of TWLog?

4.1 Experimental Setup

Dataset. We evaluate the proposed TWLog on three log datasets, HDFS, OpenStack, Kubernetes logs. HDFS (Hadoop Distributed File System) logs are from a distributed system and a high-performance computing environment [17]. We deploy a minimal OpenStack (Mitaka) environment in local which consists of a single control node, along with one network node. Additionally, we provision 3 compute nodes for the computational workload and virtual machine provisioning requirements. A test scripts is executed to generate normal and abnormal logs regularly. The tasks on scripts are VM creation/deletion, starting/stopping, adjusting VM resources (CPU, memory, disk). To simulate abnormal log entries, we create controlled failure scenarios in OpenStack such as allocate insufficient resources for VM creation.

Our experiment also refers the log data from K8S (Kubernetes) cluster which is the widely container orchestration platform [18]. We use one Linux server

to deploy K8S (v1.28.1) as Kubernetes Single-Node Setup mode which control node and worker node are running on same server. Log data is from the monitor pods which hosting applications such as InfluxDB [19], Grafana [20], and other monitoring tools deployed within Kubernetes cluster, e.g., Prometheus [21]. These monitor pods are responsible for gathering and processing various metrics, events, and logs related to the performance and its associated services. The logs were collected over a period of time during the normal operation of the Kubernetes cluster, capturing a wide range of events and activities. We simulate various abnormal scenarios by injecting anomalous events or creating failures in the monitored services. For example, send invalid queries to the InfluxDB database, resulting in error logs.

Baseline. To evaluate our proposed method TWLog, we analysis the state-of-the-art models for anomaly detection with the following baselines.

Principal Component Analysis (PCA) method [22] is technique for detecting anomalies which transforms the original high-dimensional data into a lower-dimensional space by identifying the principal components [23].

One-Class Support Vector Machine (SVM) is a machine learning algorithm used for anomaly detection in situations where only normal data is available for training [24].

DeepLog is a deep learning-based approach for log-based anomaly detection in large-scale systems [25]. It leverages recurrent neural networks (RNNs), specifically long short-term memory (LSTM) networks, to learn the temporal dependencies and patterns present in log sequences. Log sequences are represented as sequences of tokens, where each token corresponds to a specific log event or message. Then it employs LSTM networks to model the temporal dependencies within log sequences. Anomalies are identified based on deviations from the learned normal behavior.

Evaluation Metrics. The widely used metrics Precision, Recall, and F1-score are used to measure the effectiveness of anomaly detection for TWLog. Precision measures the proportion of correctly detected anomalies out of all detected anomalies. Recall measures proportion of correctly detected anomalies out of all real anomalies. F1-Score measures the harmonic mean of Precision and Recall.

4.2 RQ1: Effectiveness

To answer RQ1, we utilize a sampled dataset to figure out the effectiveness of TWLog compared to baseline approaches. The experiment conducted in three different dataset HDFS, OpenStack and Kubernetes. Results are shown in Table 1. PCA, SVM can achieve relatively good performance on dataset HDFS, their performance is extremely low in dataset OpenStack and Kubernetes. The main reason is that they use the counting vector to represent a log sequence which loss the temporal information from sequences. DeepLog perform better than PCA and SVM due to deep learning models to capture the patterns of sequences.

The performance on the public HDFS dataset is similar between DeepLog and our method. This can be attributed to the straightforward trace rela-

tions within the HDFS dataset. However, it achieves low performance in local dataset OpenStack and Kubernetes. DeepLog does not consider the trace relation among services or components. Therefore, they are likely to falsely report unseen event subsequences as anomalies. Actually, this kind of log anomaly detection approaches usually work on a single log event sequence such as the HDFS dataset [13, 26]. TWLog considers all the trace relation from business view, thus it performs better than log anomaly detection approach which is only focus the sliding window log sequence.

Table 1. Experimental results on HDFS, OpenStack, Kubernetes Datasets

Method	HDFS			OpenStack			Kubernetes		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
PCA	0.975	0.818	0.891	0.312	0.740	0.439	0.184	0.321	0.231
SVM	0.981	0.837	0.902	0.359	0.273	0.310	0.291	0.328	0.308
DeepLog	0.993	0.925	0.958	0.615	0.841	0.710	0.580	0.774	0.663
TWLog	0.991	0.923	0.954	0.956	0.973	0.963	0.931	0.894	0.909

4.3 RQ2: Impact of Relation Graph

To answer RQ2, we conduct ablation experiments on three log datasets and evaluate the performance of TWLog between only using Transformer based for log anomaly detection without constructing relation graph (no task workflow based) and Transformer based log anomaly detection with relation graph (task workflow based). Table 2 shows the experimental results. We can notice that on HDFS dataset the relation graph for task workflow has moderate effect on the performance of anomaly detection. This is because HDFS dataset exhibits relatively simple for component interactions and less complex call hierarchy. As a result, the absence of task workflows does not significantly impact the effectiveness of the anomaly detection method. However, the absence of task workflows negatively impacts the performance on the other two datasets especially for OpenStack log data. This is because the log sequences in these datasets exhibit strong business logic relationships, which are closely tied to task workflows. The performance is better for task workflow based compared to no task workflow.

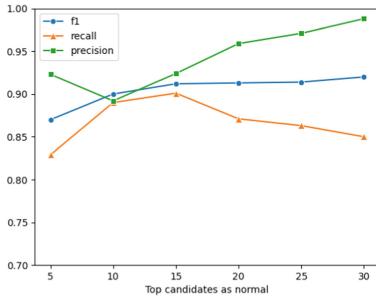
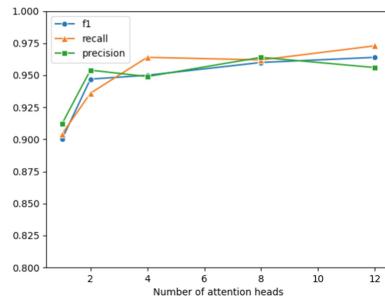
4.4 RQ3: Impact of Configurations

We adopt the OpenStack dataset to analyze the model performance by tuning hyper parameters. Figure 5 shows that the F1-Score for log anomaly detection keeps increasing with the increase the size of the candidate set of normal log keys. It is expected that the precision keeps increasing while recall keeps small decrease when the size of candidates increasing. We need to find the appropriate size of the candidate set to balance the precision and recall for the anomaly detection. Additionally, we also conduct the experiment for the number of attention heads change. From Fig. 6, we observe that the performance decreases as

Table 2. Experimental results with and without incorporating task workflow

Dataset	No task workflow basis			Task workflow basis		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
HDFS	0.983	0.915	0.948	0.991	0.923	0.954
OpenStack	0.615	0.841	0.710	0.865	0.973	0.954
Kubernetes	0.580	0.774	0.663	0.901	0.894	0.919

the number of attention heads decreases. For example, achieves F1-scores ranging from 0.950 to 0.975 when using 12 attention heads. These results are higher than those obtained by using only one attention head (0.900–0.925). In summary, the Transformer model achieves promising results with different hyperparameter values. The performance is much better when the number of attention heads is in the range of 4 and 12.

**Fig. 5.** Size of candidates.**Fig. 6.** Number of attention heads.

5 Conclusion

Log anomaly detection plays a crucial role in maintaining the reliability and security of distributed systems. In this paper, we propose a novel approach for log anomaly detection. To better capture contextual information from log sequences, we leverage self-attention mechanism and build Transformer-based log anomaly model. Moreover, task workflow-based trace relation graph is constructed. It makes that log sequences can better represent the invocation relationships of business logic distributed service systems. These sequences are then fed into the Transformer model for anomaly detection. Our experimental results on real-world datasets demonstrate the effectiveness of the proposed method. In the future, we plan to study how to design self-supervised learning tasks for anomaly detection and root cause analysis based on anomaly detection.

References

1. Guo, H., Yuan, S., Wu, X.: Logbert: log anomaly detection via bert. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
2. Opentracing. <https://opentracing.io/>
3. Lou, J.G., Fu, Q., Yang, S., Xu, Y., Li, J.: Mining invariants from console logs for system problem detection. In: USENIX Annual Technical Conference (USENIX ATC 2010), pp. 231–244. (2010)
4. Yu, X., Joshi, P., Xu, J., Jin, G., Zhang, H., Jiang, G.: Cloudseer: workflow monitoring of cloud infrastructures via interleaved logs. ACM SIGARCH Comput. Archit. News **44**(2), 489–502 (2016)
5. Zhao, X., Rodrigues, K., Luo, Y., Yuan, D., Stumm, M.: Non-intrusive performance profiling for entire software stacks based on the flow reconstruction principle. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2016), pp. 603–618 (2016)
6. He, S., Zhu, J., He, P., Lyu, M. R.: Loghub: a large collection of system log datasets towards automated log analytics. arXiv preprint [arXiv:2008.06448](https://arxiv.org/abs/2008.06448) (2020)
7. Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y.: Loganomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: IJCAI, pp. 4739–4745 (2019)
8. Wang, Z., Chen, Z., Ni, J., Liu, H., Chen, H., Tang, J.: Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3726–3734 (2021)
9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
10. Meng, W., et al.: Loganomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: IJCAI, vol. 19, no. 7, pp. 4739–4745 (2019)
11. Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y.: Robust log-based anomaly detection on unstable log data. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 807–817 (2019)
12. Wu, Y., et al.: Google’s neural machine translation system: bridging the gap between human and machine translation. arXiv preprint [arXiv:1609.08144](https://arxiv.org/abs/1609.08144) (2016)
13. Chang, H.J., Yang, S.W., Lee, H.Y.: Distilhubert: speech representation learning by layer-wise distillation of hidden-unit bert. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7087–7091 (2022)
14. Vervaet, A.: Monilog: an automated log-based anomaly detection system for cloud computing infrastructures. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 2739–2743 (2021)
15. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
16. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
17. Zhu, J., He, S., He, P., Liu, J., Lyu, M.R.: Loghub: a large collection of system log datasets for AI-driven log analytics. In: 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), pp. 355–366 (2023)

18. Kubernetes. <https://kubernetes.io/docs/home/>
19. Influxdb. <https://docs.influxdata.com/>
20. Grafana. <https://grafana.com/docs/grafana/latest/>
21. Prometheus. <https://prometheus.io/docs/introduction/overview/>
22. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.I.: Detecting large-scale system problems by mining console logs. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, pp. 117–132 (2009)
23. He, S., Zhu, J., He, P., Lyu, M.R.: Experience report: system log analysis for anomaly detection. In: 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 207–218 (2016)
24. Wang, Y., Wong, J., Miner, A.: Anomaly intrusion detection using one class SVM. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, pp. 358–364 (2004)
25. Du, M., Li, F., Zheng, G., Srikumar, V.: Deeplog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298 (2017)
26. Zhang, C., et al.: Deeptralog: trace-log combined microservice anomaly detection through graph-based deep learning. In: Proceedings of the 44th International Conference on Software Engineering, pp. 623–634 (2022)
27. He, P., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: an online log parsing approach with fixed depth tree. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 33–40 (2017)
28. Du, M., Li, F.: Spell: streaming parsing of system event logs. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 859–864 (2016)
29. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.: Largescale system problem detection by mining console logs. In: Proceedings of SOSP 2009 (2019)
30. Lin, Q., Zhang, H., Lou, J.G., Zhang, Y., Chen, X.: Log clustering based problem identification for online service systems. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp. 102–111 (2016)
31. He, S., Lin, Q., Lou, J.G., Zhang, H., Lyu, M.R., Zhang, D.: Identifying impactful service system problems via log analysis. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 60–70 (2018)
32. Zhang, K., Xu, J., Min, M.R., Jiang, G., Pelechrinis, K., Zhang, H.: Automated IT system failure prediction: a deep learning approach. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 1291–1300 (2016)
33. Zhang, X., et al.: Robust log-based anomaly detection on unstable log data. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 807–817 (2019)
34. Liu, F., Wen, Y., Zhang, D., Jiang, X., Xing, X., Meng, D.: Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 1777–1794 (2019)
35. Guo, X., et al.: Graph-based trace analysis for microservice architecture understanding and problem diagnosis. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 1387–1397 (2016)
36. Liu, P., et al.: Unsupervised detection of microservice trace anomalies through service-level deep Bayesian networks. In: 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), pp. 48–58 (2020)



TS-AUBD: A Novel Two-Stage Method for Abnormal User Behavior Detection

Yu Cao¹, Yilu Chen¹, Ye Wang^{1,3}, Ning Hu^{2,4}, Zhaoquan Gu^{1,2(✉)},
and Yan Jia^{1,2}

¹ Harbin Institute of Technology, Shenzhen 518038, China

{22S151131,23B951004}@stu.hit.edu.cn,

{wangye,guzhaoquan,jiayan2020}@hit.edu.cn

² Peng Cheng Laboratory, Shenzhen 518038, China

hun@pcl.ac.cn

³ National University of Defense Technology, Changsha 410003, China

⁴ Guangzhou University, Guangzhou 511540, China

Abstract. Malicious insider attacks are among the most destructive threats to enterprises. Solving the insider threat problem involves several challenges, including data imbalance and detection of anomalous behavior. This paper presents TS-AUBD, a two-stage method for abnormal user behavior detection. TS-AUBD consists of coarse-grained and fine-grained user-level models. TS-AUBD can not only effectively detect abnormal behaviors and users but also analyze the situation of abnormal behaviors presented in each abnormal user. Experiments were conducted on a publicly available standard dataset CERT R4.2. Results show that TS-AUBD shows better performance compared with the baseline model, with an accuracy of up to 99.9% for behavior detection and 99.8% for user detection.

Keywords: Abnormal users · Abnormal behavior · User level

1 Introduction

With the development of computer method, users' network behavior has become more and more abundant. In terms of insider threats, there is a proliferation of emerging threats. External attackers hijack the identity of employees to launch attacks, internal staff malicious intent to cause corporate information leakage, and so on, making traditional security method a huge challenge. Some abnormal behavior detection methods focus on "historical features", which have a lag and are difficult to solve new insider threats. Therefore, it is becoming necessary to research security technologies that are centered on "user behaviors" and the "behavior" as the core of the security method, which is called User and Entity Behavior Analytics (UEBA). UEBA refers to mining potential events that are different from the user's standard behavior, and the method for mining abnormal events is called abnormal user behavior detection method. UEBA enables early

detection of abnormal users from user behavior and prevents data leakage and privilege abuse within the enterprise.

Related works to UEBA's research [2-9] mainly focus on identifying abnormal behaviors and have not explored the relationship between abnormal behaviors and users. Enterprises not only pay attention to the abnormal detection of behaviors but also pay more attention to identifying abnormal users and their corresponding behaviors. In the abnormal user behavior detection scenario, there are two problems: the data imbalance problem and the high accuracy of anomalous behavior, which do not correspond to the high accuracy of anomalous users. Firstly, the data imbalance problem is extremely serious, and too few abnormal behaviors will seriously affect the model's effectiveness in detecting abnormal behaviors. Secondly, the existing studies do not judge whether the detected abnormal behaviors cover all abnormal users. Sometimes, 90% of abnormal behaviors are detected, while only correspond to 60% of abnormal users.

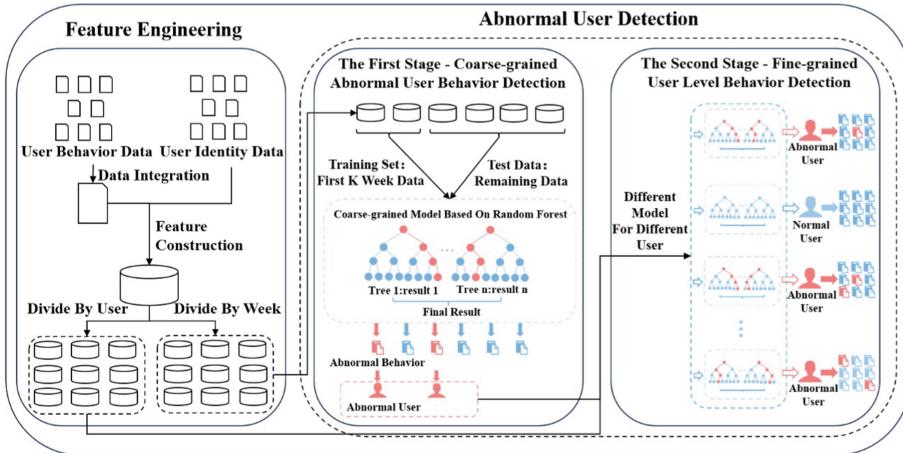


Fig. 1. Two-stage abnormal user behavior detection model

In this study, we proposed TS-AUBD, which consists of two stages of abnormal user behavior detection, and introduced data oversampling technologies to solve the data imbalance problem. As Fig. 1 shows, the first stage of TS-AUBD is a coarse-grained model that can identify abnormal users through weekly user behavior detection. The second stage of TS-AUBD is the fine-grained model, which constructs separate user-level models for each user whose anomaly is detected in the coarse-grained model. By using TS-AUBD, not only can we efficiently identify abnormal behaviors and users, but we can also obtain the behavioral detection of each abnormal user.

In conclusion, the contributions of this study are as follows:

- We propose TS-AUBD, which can chronologically detect abnormal behaviors. TS-AUBD can build user-level models for each potential abnormal user respectively, which can effectively detect abnormal users and behaviors.
- We introduced data oversampling method in TS-AUBD to balance the data and enhance the model effect.
- We evaluated TS-AUBD on the publicly available dataset CERT R4.2. Results showed that compared to the baseline method [11], TS-AUBD has superiority with an accuracy of up to 99.9% for behavior detection and 99.8% for user detection.

2 Related Work

With the continuous progress of method, the insider threat changes from unauthorized access to data theft, identity hijacking [3, 26, 27], etc., and the corresponding strategy for insider threat detection also changes from access control and user authentication to user entity behavioral analysis. In contrast, the accuracy and efficiency of anomaly detection in the user entity behavioral analysis is the key research Objective. Abnormal user behavior detection can be performed based on malicious metrics and rules [24, 28] or by constructing models based on user features using supervised learning methods or unsupervised learning methods [10–23]. Most studies performing abnormal user behavior detection are based on one model for behavioral-level abnormal behavior detection.

2.1 Studies Based on Metrics and Rules

Some researchers performed abnormal user behavior detection based on metrics or rules. Shashanka et al. [24] considered sensitive abnormal demand and variable weights to detect abnormal behavior based on singular value decomposition. Tang et al. [28] optimized the problem of identifying the problem of identifying first-time users accessing the system as abnormal alarms and the data sparsity problem by weighting different data metrics.

2.2 Studies on Behavior Model

Several researchers have constructed anomaly detection models using different supervised and unsupervised learning methods to process the data. Suresh et al. [29] used a fuzzy affiliation function for feature aggregation and RWMA for converting the traditional Random Forest into a perceptron-like algorithm to detect abnormal user behaviors. Singh et al. [25] use Bi-LSTM for feature extraction and SVM for abnormal behaviors. Le et al. [11] proposed a user-centric insider threat detection system that analyses data at multiple data granularity levels under different training conditions. Al-Mhiqani et al. [1] proposed a multi-layer insider

threat detection framework that selects the most appropriate model from multiple insider threat detection classification models based on multi-criteria decision-making methods. Pantelidis et al. [22] use Variational Autoencoder Deep learning algorithms for automatic defense against insider threats.

However, these studies did not correspond the behaviors to the anomalies of the users, and the generation of behaviors in the real situation is temporal, which needs to be detected according to the generation pattern of the behavioral data instead of just dividing the overall data proportionally to the training set. To solve these problems, this paper proposes a two-stage abnormal user behavior detection model TS-AUBD.

3 Method

This research uses the Random Forest(RF) algorithm to detect abnormal users and behaviors through historical user behavioral data. Experiments have been conducted in this study, and it is found that random forest performs better than other algorithms in the abnormal user behavior detection scenario. As an integrated algorithm, random forest itself can build multiple decision trees to learn high dimensional features in user behavior data, and can resist the impact of data imbalance. It corresponds abnormal users to abnormalities in behaviors through a two-stage model to accurately analyze users and their behaviors to help companies and organizations improve their insider threat prevention capabilities. In this paper, the description will be developed according to the following steps:

- **Feature Engineering:** Integrate behavioral data from different user activities and perform feature extraction according to behaviors and subset data division after feature processing.
- **Model Construction:** The abnormal user and behavior detection model is divided into two stages. The first stage of TS-AUBD establishes a coarse-fined model and chronologically detects user behaviors. In the second stage of TS-AUBD, the Random Forest algorithm is used to construct an independent user abnormal behavior detection model for each malicious user.
- **Result Analysis:** The detection results of the model are analyzed through the confusion matrix, accuracy rate, false positive rate, etc.

3.1 Problem Definition

In the scenario of anomaly user behavior detection, user behavior data can be obtained and we denote the behavior set as $Behaviors = \{b_1, b_2, \dots, b_n\}$. And we denote the user set as $Users = \{u_1, u_2, \dots, u_m\}$, the goal is to detect abnormal behavior, or the User. We can define this problem as:

Problem 1: There is some abnormal user behavior data in the user behavior data need be detected. The problem is to detect abnormal user behaviors as more as possible, input user behavior data into detection model to determine whether each behavior is abnormal. The model determines that normal behavior is 1,

abnormal behavior is 0, and the set composed of abnormal behavior is denoted as $AbnormalBehaviors = \{b_a1, b_a2, \dots b_ap\}$.

Problem 2: When the abnormal user behavior detection is carried out, the abnormal user detection is also a very important part. Therefore, it is necessary to find the abnormal users from all users as much as possible, a user with abnormal behavior is an abnormal user, we can denote the abnormal users set as $AbnormalUsers = \{u_a1, u_a2, \dots u_aq\}$.

3.2 Feature Engineering

Feature engineering is the process of transforming raw data into features that can more accurately express the nature of the problem. In this paper, feature engineering refers to the extraction of more features from the integrated user behavior and identity information, the extraction of features partially refers to the study of Le [11]. Feature engineering helps TS-AUBD to detect abnormal users and abnormal behaviors. Feature engineering, including integrating user behavior data and feature construction.

- Usually, the organizational information in the data will contain two parts: user behavioral information and user identity information. User behavioral information will contain a variety of user behavioral activity logs, such as user internet behavior logs, user USB usage logs, etc., each of which is described by different attributes. All activity logs must be integrated to analyze the user's behavioral patterns.
- Constructing different features for different behaviors. For example, behaviors involving email sending information can be related to features like “the number of recipients,” “the email length,” etc.
- User identity and behavioral information can be combined to construct features, such as user behavior time risk rating.
- After completing the construction of features, the data is encoded with natural numbers. Time features such as days and weeks can be extracted based on the time of the behavior. The raw data is transformed through feature processing into a format that the model can use, providing the model with more features to analyze user behavior.
- This research aims to analyze user behavioral habits from their historical behavioral data to detect anomalies. In the first stage of TS-AUBD, we can simulate realistic chronological detection of anomalous behaviors and users. In the second stage of TS-AUBD, we can construct separate models for each user to simulate the behavioral differences between different users, to detect possible anomalous behaviors for different users better. We split the encoded data by week and user, to achieve this goal.

3.3 Stage 1: Coarse-Grained Abnormal Behavior Detection

The first stage of TS-AUBD requires establishing a coarse-grained model and chronologically detecting user behaviors every week to identify abnormal users. The detection in the first stage includes the following parts:

- Analyse the time of all user behaviors and the number of weeks in which the behaviors start and end. To ensure no users end activities before detection starts, the number of weeks of data that make up the training set needs to be lower than the earliest week the user ends all behaviors. Assuming that the user who ended all behaviors earliest ended in the K week. So, the behavioral data belonging to the $[0, K-5]$ weeks is used as the model training set, and the training set data is oversampled.
- The model was constructed by random forest algorithm. Input the remaining data chronologically by week as the test set to detect user behavior by week.
- When abnormal user behavior is detected, the user is immediately handed over to the second stage, a fine-grained model for user-level abnormal behavior analysis and detection of the user.

3.4 Stage 2: Fine-Grained Abnormal Behavior Detection

The second stage is fine-grained user-level behavior detection, which further detects users with anomalies identified by the coarse-grained model. This process consists of data oversampling and user-level anomalous behavior detection. The process is as follows:

- The user's behavioral data to be detected is randomly divided into a training set and test set according to 7:3, and the training set is subjected to data oversampling. If no data is labeled as anomalous behavior in the training set, we won't oversample the training set. This is in consideration of the fact that the abnormal users identified by the coarse-grained model in the first stage may be misdiagnosed, and the normal users who are misdiagnosed as abnormal users naturally do not have abnormal behaviors in their behavioral data and, therefore, do not have abnormal behaviors in their training set.
- Since different users have different behavioral habits, constructing a separate user behavior model for each user helps to effectively detect their abnormal behavior and reduce false positives for normal behavior. Considering the multidimensional features and unbalanced nature of data, we chose Random Forest for model construction to train a separate model for each user and detect it. Random forest models can handle high-dimensional and complex data and are suitable for multi-dimensional user entity behavior analysis features. Finally, the model performance is tested, and several performance metrics are used to evaluate the detection of the model.

In summary, TS-AUBD can solve the problems in abnormal user and behavior detection scenarios. TS-AUBD can effectively detect abnormal users and behaviors and analyze their corresponding abnormal behavioral situations.

4 Experiment

We evaluated TS-AUBD on a public benchmark dataset and show the results in this section. We first introduce the setup details and then analyze the experimental results.

4.1 Dataset

This study uses the R4.2 dataset from CMU-CERT, a highly recognized insider threat dataset in academia, to explore the research questions in this paper. Given that the focus of this study is to construct a separate anomalous behavior detection model for each user and to train and test the effectiveness of the model by using the anomalous user data, the R4.2 dataset in CMU-CERT is chosen as the experimental dataset, which contains a larger number of anomalous users and can better test the effectiveness of the model. The CMU-CERT dataset contains enterprise user behavioral data, where a scenario with 1000 employees is simulated and 70 of them are anomalous employees. The dataset contains several data files, including logon, device, HTTP, email, file, and administrative records in LDAP folders. The logon registration behavior, flash drive behavior, network access behavior, email behavior, and file copying behavior using flash drive are recorded. The administrative records in the LDAP folder are used to record the active employee list for each month. If an employee leaves or is terminated from the activity, the user will no longer be included in the employee list for that month.

4.2 Experimental Setup

This study aims to detect abnormal user behavior. We combine model training and testing using user behavior data and user identity information from the CERT dataset to evaluate the system's performance. Abnormal behaviors that are identified as normal behaviors or normal behaviors that are identified as abnormal behaviors are considered to be misclassified at the behavioral level. In this study, a user with one abnormal behavior is considered abnormal. The test result is a binary classification problem that is divided into two categories: normal behavior and abnormal behavior. In cyber security applications of machine learning, confusion matrices can be used to determine the specific classification in normal and abnormal behavior classification tasks. In this study, a binary classification problem, the confusion matrix is a 2*2 matrix with True Positive (normal behavior that is correctly classified), False Negative (normal behavior that is misclassified as abnormal behavior), False Positive (abnormal behavior that is misclassified as normal behavior) and True Negative (correctly Classified Abnormal behavior). The performance metrics include five commonly used performance metrics as follows:

- Accuracy (ACC): Proportion of behaviors predicted correctly

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision (PRE): Proportion of correctly predicted normal behaviors to total predicted normal behaviors.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall (DR): Proportion of correctly predicted normal behaviors to total normal behaviors.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1 score (F1): Reflects combined consideration of precision and recall of model predictions

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

- FPR: The proportion of abnormal behaviors that are falsely detected as normal behaviors measures the model's ability to discriminate between samples of negative instances.

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

4.3 Data Oversampling Experiment

The proportion of abnormal behaviors is very low, both among overall behaviors and abnormal users, which may seriously affect the model's performance and hinder its detection of abnormal behaviors.

Table 1. Range of abnormal user behavior

Metrics	Range
Proportions of abnormal behavior in a single abnormal user	0.02%–6.77%
Number of abnormal behaviors in a single abnormal user	5–350
Number of behaviors in a single abnormal user	2,115–56,452

Table 2. Abnormal behavioral situations

Metrics	Anomalies	Totals	Proportions of anomalies
All Users Situation	70	999	7.01%
All Behavioral Situations	7,273	32,770,222	0.02%
Abnormal User A behavior Situation	10	45,192	0.02%
Abnormal User B's behavioral Situation	5	3,035	0.16%

Table 3. Abnormal users with different proportions of abnormal behavior

Metrics	Proportions
The proportion of users with less than 0.1% of abnormal behavior as a proportion of abnormal users	40.0%
The proportion of users with less than 1% of abnormal behavior as a proportion of abnormal users	78.6%
The proportion of users with less than 2% of abnormal behavior as a proportion of abnormal users	85.7%

As shown in Table 1, the proportion of anomalous behavior among abnormal users is only 0.02% to 7.26%. As shown in Table 2, the proportion of abnormal behaviors in the total behaviors in the dataset is only 0.02%. As shown in Fig. 2 and Table 3, 40% of the abnormal users have an abnormal behavior of less than 0.1%, and more than 78% of the abnormal users have abnormal behavior proportion of less than 1%. The same is true for the proportions of abnormal behaviors in the real business.

To verify whether data oversampling can optimize the model effect, the over-sampled training set and the original training set can be fed separately into the model for training.

Using the Random Forest and XGBoost models in over-sampling experiments, the effect is shown in Table 4. Results can be seen, after the data over-sampling, the precision of the two models has been improved. Before and after data oversampling, the recall did not change, the FPR declined. It indicates that before and after data oversampling, the misclassification of abnormal behaviors in the test set is reduced, so data oversampling can solve the problem of insufficient abnormal behaviors and enhance the model's detection effect.

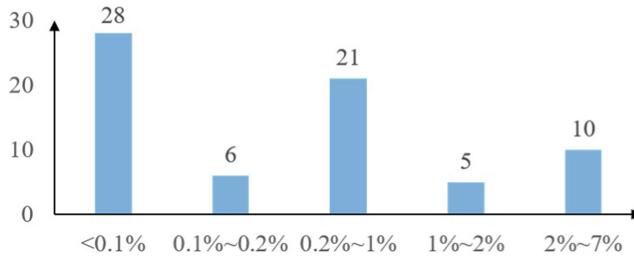


Fig. 2. Distribution of abnormal behavior among abnormal users

Table 4. Model effects before and after data oversampling

Model	ACC	PRE	DR	F1	FPR
Random Forest, not oversampled	0.9997	0.9997	1.0	0.9998	0.8587
Random Forest, oversampled	0.9998	0.9998	1.0	0.9998	0.7576
XGBoost, not oversampled	0.9998	0.9998	1.0	0.9999	0.8976
XGBoost, oversampled	0.9998	0.9998	1.0	0.9999	0.7724

4.4 Comparative Experiment

In this paper, we trained TS-AUBD by historical user behavior data. In the first stage of TS-AUBD, the coarse-grained model chronologically identifies users that may be anomalous, and the identified anomalous users are handed over to the fine-grained user-level model in the second stage for detection. In the second stage of TS-AUBD, the model is trained separately for each user.

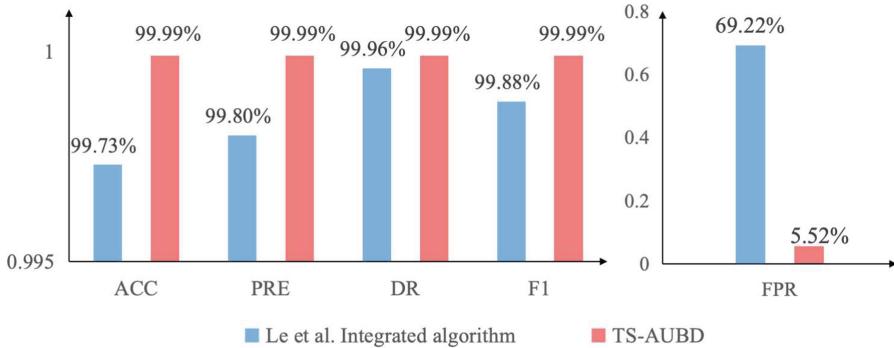


Fig. 3. Comparative experiment results

Integrated algorithm [11] by Le et al. is a state-of-the-art method, so this algorithm is used as a baseline model to compare with TS-AUBD. Experiment results can be seen in Fig. 3, both TS-AUBD and baseline model have higher accuracy because most of the behaviors are normal behaviors. High accuracy can be achieved by correctly classifying most of the normal behaviors. From the precision, recall, and F1 score, TS-AUBD performs better. It shows that TS-AUBD can correctly detect more normal behaviors and abnormal behaviors. TS-AUBD can reduce misclassified behavior. TS-AUBD has a lower FPR, which indicates that TS-AUBD has a better ability to identify abnormal behaviors.

Table 5. Confusion matrix for user detection situations

	Positive	Nagetive
Positive	930	0
Nagetive	4	66

Most existing methods for detecting abnormal user behavior focus on detecting behaviors, while TS-AUBD can detect abnormal behaviors and abnormal users. The detection of users by TS-AUBD is shown in Table 5. Two abnormal users are not detected, and all other normal and abnormal users are correctly classified. As shown in Fig. 4, TS-AUBD can detect users with an accuracy of 99.98%, while the false positive rate is only 5.71%.

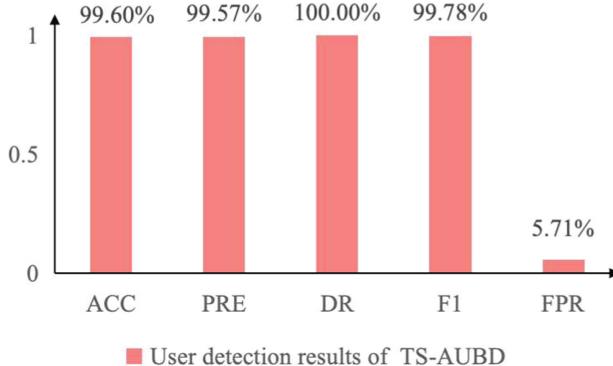


Fig. 4. User detection results of TS-AUBD

4.5 Ablation Study

The two-stage model in this paper can be divided into two parts. The coarse-grained model is in the first stage, and the fine-grained user-level model is in the second stage. In order to show the effect of different parts of TS-AUBD, the coarse-grained model is evaluated. The results are shown in Table 6, and the detection of users is shown in Table 7.

Table 6. Ablation of experimental behavioral levels

Model	ACC	PRE	DR	F1	FPR
Stage 1 of TS-AUBD	0.9996	0.9996	0.9999	0.9998	0.9777
TS-AUBD	0.9999	0.9999	0.9999	0.9999	0.0552

Interestingly, Table 6 shows that anomalous user behavior detection performance in the first stage is slightly less than that of TS-AUBD. The FPR of 0.9777 in the coarse-grained model for behavior detection is much higher than the FPR of 0.0552 in TS-AUBD. It suggests that the coarse-grained anomalous user behavior detection model in the first stage alone causes many anomalous behaviors to be misclassified. This misclassification problem can be solved by the fine-grained user-level abnormal behavior detection model in the second stage. Table 7 shows that compared to the coarse-grained model, TS-AUBD has improved in accuracy, precision, and recall, with the recall reaching 1.0. This situation indicates that the coarse-grained model in the first stage misclassified some normal users. In contrast, the second-stage model correctly classified these normal users with further detection. From Table 7, there is no change in the FPR in the overall detection results of the first-stage model and TS-AUBD, which means that the first-stage model misclassifies some abnormal users and the second-stage model can't correctly classify these users. Since the first-stage

model did not identify that small portion of abnormal users as abnormal, it was not handed over to the second-stage model for further detection, so these portions of abnormal users couldn't be correctly classified after the second-stage model.

Table 7. Ablation of experimental user levels

Model	ACC	PRE	DR	F1	FPR
Stage 1 of TS-AUBD	0.9310	0.9954	0.9301	0.9616	0.0571
TS-AUBD	0.9960	0.9957	1.0	0.9978	0.0571

These experiment results may imply some relationship between the two stages of the model, where the first-stage model initially classifies the user, and the detection effect on the behavior is not precise. In contrast, the second-stage model further identifies the user and the behavior, greatly reducing misclassification and improving the model effect. It can be seen that this paper not only manages to correspond to the detection of abnormal behavior in each user but also achieves better detection results. Our method provides an effective solution for detecting abnormal user behavior, which is of theoretical significance in helping enterprises detect abnormal behavior and users.

5 Conclusions

This paper proposes a two-stage abnormal user behavior detection method that can effectively detect abnormal users and abnormal behaviors. TS-AUBD can effectively identify normal behaviors and abnormal behaviors. TS-AUBD can also effectively identify normal and abnormal users. Due to the fine-grained user-level behavioral model in the second stage, TS-AUBD can also correspond to the detection of abnormal behavior in abnormal users. Experiment on the CERT R4.2 dataset shows that TS-AUBD can achieve better behavioral detection effect and user detection effects compared to baseline model.

Acknowledgments. This work was supported in part by the Shenzhen Science and Technology Program (No. KJZD20231023094701003), the Major Key Project of PCL (Grant No. PCL2023A07-4), and the National Natural Science Foundation of China (Grant No. 62372137).

References

1. Al-Mhiqani, M.N., et al.: A new intelligent multilayer framework for insider threat detection. *Comput. Electr. Eng.* **97**, 107597 (2022)
2. Aldairi, M., Karimi, L., Joshi, J.: A trust aware unsupervised learning approach for insider threat detection. In: 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), pp. 89–98 (2019)

3. AlSlaiman, M., Salman, M.I., Saleh, M.M., Wang, B.: Enhancing false negative and positive rates for efficient insider threat detection. *Comput. Secur.* **126**, 103066 (2023)
4. Besnaci, S., Hafidi, M., Lamia, M.: Dealing with extremely unbalanced data and detecting insider threats with deep neural networks. In: 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAEECS), pp. 1–6 (2023)
5. Ge, D., Zhong, S., Chen, K.: Multi-source data fusion for insider threat detection using residual networks. In: 2022 3rd International Conference on Electronic Information and Communication Technology (CECIT), pp. 359–366 (2022)
6. Hall, A.J., Pitropakis, N., Buchanan, W.J., Moradpoor, N.: Predicting malicious insider threat scenarios using organizational data and a heterogeneous stack-classifier. In: 2018 IEEE International Conference on Big Data Big Data, pp. 5034–5039 (2018)
7. He, W., Wu, X., Wu, J., Xie, X., Qiu, L., Sun, L.: Insider threat detection based on user historical behavior and attention mechanism. In: 2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC), pp. 564–569 (2021)
8. Huang, W., Zhu, H., Li, C., Lv, Q., Wang, Y., Yang, H.: ITDBERT: temporal-semantic Representation for Insider Threat Detection. In: 2021 IEEE Symposium on Computers and Communications (ISCC), pp. 1–7 (2021)
9. Igbe, O., Saadawi, T.: Insider threat detection using an artificial immune system algorithm. In: 2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), pp. 297–302 (2018)
10. Jah Rizvi, S.K., Javed, K.F., Moazam, M.: CAS - attention based ISO/IEC 15408-2 compliant continuous audit system for insider threat detection. In: 2023 3rd International Conference on Artificial Intelligence (ICAI), pp. 153–157 (2023)
11. Le, D.C., Zincir-Heywood, N., Heywood, M.I.: Analyzing data granularity levels for insider threat detection using machine learning. *IEEE Trans. Netw. Serv. Manag.* **17**(1), 30–44 (2020)
12. Lin, L., Zhong, S., Jia, C., Chen, K.: Insider threat detection based on deep belief network feature representation. In: 2017 International Conference on Green Informatics (ICGI), pp. 54–59 (2017)
13. Liu, A., Du, X., Wang, N.: Recognition of access control role based on convolutional neural network. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 2069–2074 (2018)
14. Mamidanna, S.K., Reddy, C.R.K., Guju, A.: Detecting an insider threat and analysis of XGBoost using hyperparameter tuning. In: 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), pp. 1–10 (2022)
15. Mehmood, M., Amin, R., Muslam, M.M.A., Xie, J., Aldabbas, H.: Privilege escalation attack detection and mitigation in cloud using machine learning. *IEEE Access* **11**, 46561–46576 (2023)
16. Meng, F., Lou, F., Fu, Y., Tian, Z.: Deep learning based attribute classification insider threat detection for data security. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), pp. 576–581 (2018)
17. Meng, F., Lu, P., Li, J., Hu, T., Yin, M., Lou, F.: GRU and multi-autoencoder based insider threat detection for cyber security. In: 2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC), pp. 203–210 (2021)
18. Mittal, A., Garg, U.: Design and analysis of insider threat detection and prediction system using machine learning techniques. In: 2023 Fifth International Conference

- on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–8 (2023)
- 19. Mittal, A., Garg, U.: Prediction and detection of insider threat detection using emails: a comparision. In: 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), pp. 1–6 (2023)
 - 20. Nasir, R., Afzal, M., Latif, R., Iqbal, W.: Behavioral based insider threat detection using deep learning. *IEEE Access* **9**, 143266–143274 (2021)
 - 21. Orizio, R., Vuppala, S., Basagiannis, S., Provan, G.: Towards an explainable approach for insider threat detection: constraint network learning. In: 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), pp. 42–49 (2020)
 - 22. Pantelidis, E., Bendiab, G., Shiaeles, S., Kolokotronis, N.: Insider threat detection using deep autoencoder and variational autoencoder neural networks. In: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 129–134 (2021)
 - 23. Saaudi, A., Al-Ibadi, Z., Tong, Y., Farkas, C.: Insider threats detection using CNN-LSTM model. In: 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 94–99 (2018)
 - 24. Shashanka, M., Shen, M.Y., Wang, J.: User and entity behavior analytics for enterprise security. In: 2016 IEEE International Conference Big Data Big Data, Washington DC, USA, pp. 1867–1874. IEEE (2016)
 - 25. Singh, M., Mehtre, B.M., Sangeetha, S.: User behaviour based insider threat detection in critical infrastructures. In: 2021 2nd International Conference on Secure Cyber Computing and Communication (ICSCCC), pp. 489–494 (2021)
 - 26. Sun, D., Liu, M., Li, M., Shi, Z., Liu, P., Wang, X.: DeepMIT: a novel malicious insider threat detection framework based on recurrent neural network. In: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 335–341 (2021)
 - 27. Sun, X., Wang, Y., Shi, Z.: Insider threat detection using an unsupervised learning method: COPOD. In: 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), pp. 749–754 (2021)
 - 28. Tang, B., Hu, Q., Lin, D.: Reducing false positives of user-to-entity first-access alerts for user behavior analytics. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 804–811 (2017)
 - 29. Varsha Suresh, P., Lalitha Madhavu, M.: Insider attack: internal cyber attack detection using machine learning. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1–7. IEEE (2021)



Multi-label Out-of-Distribution Detection with Spectral Normalized Joint Energy

Yihan Mei¹ , Xinyu Wang¹ , Dell Zhang² , and Xiaoling Wang¹

¹ East China Normal University, Shanghai 200062, China

`{yihanmei,xinyu_wang}@stu.ecnu.edu.cn, xlwang@cs.ecnu.edu.cn`

² Institute of Artificial Intelligence (TeleAI), China Telecom Corp Ltd.,
Beijing, China
`dell.z@ieee.org`

Abstract. In today’s interconnected world, achieving reliable out-of-distribution (OOD) detection poses a significant challenge for machine learning models. While numerous studies have introduced improved approaches for multi-class OOD detection tasks, the investigation into *multi-label* OOD detection tasks has been notably limited. We introduce Spectral Normalized Joint Energy (SNoJoE), a method that consolidates label-specific information across multiple labels through the theoretically justified concept of an energy-based function. Throughout the training process, we employ *spectral normalization* to manage the model’s feature space, thereby enhancing model efficacy and generalization, in addition to bolstering robustness. Our findings indicate that the application of spectral normalization to *joint energy* scores notably amplifies the model’s capability for OOD detection. We perform OOD detection experiments utilizing PASCAL-VOC as the in-distribution dataset and ImageNet-22K or Texture as the out-of-distribution datasets. Our experimental results reveal that, in comparison to prior top performances, SNoJoE achieves 11% and 54% relative reductions in FPR95 on the respective OOD datasets, thereby defining the new *state of the art* in this field of study.

Keywords: OOD Detection · Multi-label Classification · Spectral Normalization

1 Introduction

In the current digital era, the pervasive use of machine learning models is undeniable. However, these models often grapple with data that deviates from their training data, known as out-of-distribution (OOD) data, when deployed in real-world settings. This discrepancy can lead to inaccurate predictions, raising safety concerns and other issues. OOD detection plays a crucial role in identifying unfamiliar data, thereby enhancing model safety and robustness in diverse environments. Thus, assessing OOD uncertainty emerges as a critical challenge for researchers.

Significant advancements have been made in OOD detection research. The Local Outlier Factor (LOF) method [6] and unsupervised outlier detection using globally optimal sample-based Gaussian Mixture Models (GMM) by Yang et al. [45] represent foundational work. G-ODIN [20] builds on ODIN [24] to improve sensitivity to covariate shifts. OpenMax [5] introduces Extreme Value Theory (EVT) to neural networks, calibrating logits with EVT probability models, including the Weibull distribution. Classification-based approaches see innovations like extending One-Class Classification (OCC) through elastic-net regularization for learning decision boundaries [33], and selecting reliable data from unlabeled sources as negative samples for supervised anomaly detection settings [8].

Despite these advancements, OOD detection in multi-label classification contexts remains underexplored. Multi-label classification poses unique challenges due to the necessity of evaluating uncertainty across multiple labels, rather than a single dominant one [39]. Achieving stable model training is essential for accurate multi-label OOD sample identification, with strategies like using free energy for OOD uncertainty assessment proposed by Liu et al. [28].

This paper introduces a novel approach, **Spectral Normalized Joint Energy** (SNoJoE), for assessing OOD uncertainty in multi-label datasets. SNoJoE calculates free energy for each label and combines these energies, overcoming the difficulties generative models face in estimating joint likelihood for multi-label data [19]. Additionally, it demonstrates that aggregating label energies is more effective than summing label scores in OOD detection evaluations [39], highlighting the importance of choosing the right label assessment function.

We also utilize ResNet for feature extraction from in-distribution images, employing an energy function as the metric for OOD assessment. To counter overfitting and enhance model robustness, we apply spectral normalization as a regularization technique. Our findings show that spectral normalization reduces gradient variation ranges during training, minimizing the risk of gradient problems and promoting a well-regulated feature space. This approach helps the model to generalize better to OOD instances by focusing on extracting generalizable features rather than memorizing training data. Applying spectral normalization to OOD detection tasks has been shown to significantly improve model performance, such as achieving a significant 54% reduction in FPR95 on the Texture dataset with respect to PASCAL-VOC (t -test p -value < 0.01), underscoring the technique’s value in OOD detection.

Our main contributions include:

- Introducing SNoJoE, an innovative method for OOD uncertainty assessment in multi-label classification that can deliver today’s best performance on two real-world datasets.
- Demonstrating through ablation studies that spectral normalization significantly enhances multi-label OOD detection performance.
- Making our experimental code and datasets available for reproducible research¹.

¹ https://github.com/Nicholas-Mei/Ood_Detection_SNoJoE.

2 Related Work

2.1 Multi-label Classification

Unlike the simpler scenario of multi-class (single-label) classification, multi-label classification allows each image to be associated with multiple label concepts. Early approaches to multi-label classification treated the presence of each label independently, neglecting the potential correlations among labels [15, 43].

Initial research in multi-label classification demanded significant computational resources. Ghamrawi and McCallum [14] employed Conditional Random Fields (CRF) to create graphical models that identify correlations between labels, and Chen et al. [9] integrated CRF with deep learning techniques to examine the dependencies among output variables. These strategies necessitate the explicit modeling of label correlations, leading to elevated computational demands.

Conversely, deep learning techniques do not inherently require substantial computational resources for multi-class recognition tasks and have shown notable effects [37, 40]. Gong et al. [15] utilized Convolutional Neural Networks (CNN) to label images with 3 or 5 labels in the NUS-WIDE dataset, while Chen et al. [10] applied CNNs to categorize road scene images from a set of 52 potential labels. Thus, efficiently solving multi-label classification challenges is intricately linked to a wide range of applications in the contemporary open world.

2.2 Out-of-Distribution Detection

In the realm of OOD detection, research has primarily concentrated on four areas: Novelty Detection (ND), Open Set Recognition (OSR), Outlier Detection (OD), and Anomaly Detection (AD).

Initially, methods leaned heavily on confidence estimation and the setting of thresholds, judging inputs' relevance to known categories by the confidence scores produced by the model. However, they often falter when facing complex data distributions. Zhang et al. [47] introduced OpenHybrid, a strategy that combines representation space learning from both an inlier classifier and a density estimator, the latter acting as an outlier detector.

Ayadi et al. [2] outlined twelve diverse interpretations of outliers, highlighting the challenge of defining outliers precisely. This has spurred a wave of innovative approaches for identifying and addressing outliers [31]. Among them, density-based methods for detecting outliers represent some of the earliest strategies. The Local Outlier Factor (LOF) method, introduced by Breunig et al. [6], stands as a pioneering density-based clustering technique for outlier detection, leveraging the concept of loose correlation through k-nearest neighbors (KNN). LOF calculates local reachability density (LRD) within each point's KNN set and compares it to the densities of neighbors within that set.

Yang et al. [45] proposed an unsupervised outlier detection approach using a globally optimal sample-based Gaussian Mixture Model (GMM), employing the Expectation-Maximization (EM) algorithm for optimal fitting to the dataset.

They define an outlier factor for each data point as the weighted sum of mixture proportions, where weights denote the relationships among data points.

Certain studies have focused on increasing sensitivity to covariate shifts by examining hidden representations in neural networks' intermediate layers. Generalized ODIN [20] builds on ODIN [24] by adopting a specialized training objective, DeConf-C, and choosing hyperparameters like perturbation magnitude for in-distribution data. Wei et al. [42] demonstrated that issues of overconfidence could be alleviated through Logit Normalization (Logit Norm), which counters the typical cross-entropy loss by enforcing a constant vector norm on logits during training, enabling neural networks to distinctly differentiate between in-distribution and OOD data. Other efforts have sought to refine OOD uncertainty estimation via confidence scores based on Mahalanobis distance [23] and gradient-based GradNorm scores [21].

Within classification-based OOD detection methods, One-Class Classification (OCC) uniquely establishes a decision boundary matching the expected normal data distribution density level set [36]. Deep SVDD [34] was the first to adapt classical OCC for deep learning, mapping normal samples to a hypersphere to delineate normality. Deviations from this model are flagged as anomalous. Later efforts expanded this approach through elastic regularization [33] or adaptive descriptions with multi-linear hyperplanes [41]. Additionally, some methods employ Positive-Unlabeled (PU) learning in semi-supervised AD contexts, providing unlabeled data alongside normal data. Mainstream PU strategies either select reliable negative samples for a supervised AD setting, using clustering [8] and density models [16], or treat all unlabeled data as noise negatives for learning with noise labels, employing sample re-weighting [29] and label cleaning [35, 49].

Despite advancements, OOD detection remains a challenging field, predominantly explored within multi-class tasks, with limited work in multi-label classification. Hence, we introduce a technique that integrates spectral normalization into the network and utilizes energy scores to derive label-wise joint energy scores for OOD detection tasks.

2.3 Energy-Based Models

Energy-based models (EBMs) in machine learning trace their origins to Boltzmann machines [1]. This approach offers a cohesive framework encompassing a broad spectrum of learning algorithms, both probabilistic and deterministic [22, 32]. Xie et al. [44] showed that the discriminative classifiers within GAN networks can be interpreted through an energy-based lens. Moreover, these methods have been leveraged for structured prediction challenges [38].

Recent studies [25, 28] have advocated for the use of energy scores in detecting OOD instances, grounding their arguments in theoretical perspectives related to likelihood [30]. Here, samples exhibiting lower energy are classified as in-distribution (ID), while those with higher energy are flagged as OOD. Liu et al. [28] pioneered a technique for quantifying OOD uncertainty by utilizing energy scores, showcasing remarkable efficacy in multi-class classification networks. Meanwhile, research by Wang et al. [39] targets multi-label contexts,

illustrating the benefits of harnessing the collective power of all label data. Our contribution merges cross-label energy scores, affirming enhanced performance through the implementation of spectral normalization.

3 Method

In this section, we introduce a novel approach for OOD detection in multi-label scenarios. First, we address multi-label inputs by integrating concepts from the free energy function, assessing OOD uncertainty through the evaluation of joint label energies across labels. Subsequently, we present SNoJoE, a technique that applies spectral normalization to the joint label energy scores. This enhancement not only improves the model's robustness but also facilitates the extraction of features that are more generalizable.

3.1 Preliminaries

Multi-label Classification. Multi-label classification is a machine learning task where the goal is to assign input data samples to one or more categories out of a set of predefined labels. Unlike traditional single-label classification tasks, where each sample can only belong to one category, multi-label classification allows a sample to have multiple labels simultaneously. Generally, consider \mathcal{X} (representing the input space) and \mathcal{Y} (representing the output space), with \mathcal{P} denoting a distribution over $\mathcal{X} \times \mathcal{Y}$. Suppose $f : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Y}|}$ represents a neural network trained on samples drawn from \mathcal{P} . Each input can be correlated with a subset of labels in $\mathcal{Y} = 1, 2, \dots, K$, denoted by a vector $\mathbf{y} = [y_1, y_2, \dots, y_K]$, where

$$y_i = \begin{cases} 1, & \text{if } i \text{ is associated with } x \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Utilizing a convolutional neural network (CNN) with a shared feature space, we generate multi-label output predictions. This approach has emerged as the standard training mechanism for multi-label classification tasks, finding widespread application across various domains [27, 48].

Out-of-Distribution Detection. Similar to the concept presented in [39], we define the problem of OOD detection for multi-label classification as follows. Let \mathcal{D}_{in} denote the marginal distribution \mathcal{P} over the label set \mathcal{X} , representing the distribution of in-distribution data. During testing, the environment may generate out-of-distribution data \mathcal{D}_{out} on \mathcal{X} . The goal of OOD detection is to define a decision function D such that:

$$D(x; f) = \begin{cases} 1, & \text{if } x \sim \mathcal{D}_{in} \\ 0, & \text{if } x \sim \mathcal{D}_{out} \end{cases} \quad (2)$$

Energy Function. The definition of the energy equation was first proposed by Liu et al. They introduced the free energy as the scoring function for OOD uncertainty assessment in a multi-class setting. Given a classifier $f(\mathbf{x}) : \mathcal{X} \rightarrow \mathbb{R}^K$ mapping the input x to K real numbers as logits, the class distribution is represented through softmax:

$$p(y_i = 1 | x) = \frac{e^{f_{y_i}(x)}}{\sum_{j=1}^K e^{f_{y_j}(x)}}. \quad (3)$$

Then, the transformation from logits to probability distribution is achieved through the Boltzmann distribution:

$$p(y_i = 1 | x) = \frac{e^{-E(x, y_i)}}{\int_{y'} e^{-E(x, y')} dy'} = \frac{e^{-E(x, y_i)}}{e^{-E(x)}}. \quad (4)$$

Thus, the initially defined classifier can be interpreted from an energy-based perspective. Viewing the logits $f_{y_i}(x)$ as an energy function $E(x, y_i)$, we can obtain the free energy function $E(x)$ for any given input x :

$$E(x) = -\log \sum_{i=1}^K e^{f_{y_i}(x)}. \quad (5)$$

3.2 Label-Wise Joint Energy

We first consider the problem of OOD uncertainty detection on a standard multi-label classifier. For a given input x , its output for the i -th class is:

$$f_{y_i}(x) = h_{l-1}(x) \cdot w_{cls}^i, \quad (6)$$

where $h_{l-1}(x)$ is the feature vector of the penultimate layer of the network, and w_{cls}^i is the weight matrix corresponding to i -th class. The predictive probability of label y_i is then implemented through a variant of a binary logistic classifier:

$$p(y_i = 1 | x) = \frac{e^{f_{y_i}(x)}}{1 + e^{f_{y_i}(x)}}. \quad (7)$$

For the logistic form in Eq. 7, we can consider it as a softmax form with only 0 and $e^{f_{y_i}(x)}$ as the logits:

$$p(y_i = 1 | x) = \frac{e^{f_{y_i}(x)}}{e^0 + e^{f_{y_i}(x)}}. \quad (8)$$

Through the softmax form of Eq. 8, for each $i \in \{1, 2, \dots, K\}$, the *energy function* of class y_i can be expressed as follows:

$$E_{y_i}(x) = -\ln(1 + e^{f_{y_i}(x)}). \quad (9)$$

Therefore, for each class $\{y_i\}_{i=1}^K$, we can derive a *label-wise joint energy function* as follows:

$$E_{joint}(x) = \sum_{i=1}^K -E_{y_i}(x) \quad (10)$$

In Eq. (9), we consider the joint uncertainty among labels. Wang et al. [39] provided a theoretical foundation based on joint likelihood. Subsequent work by Zhang and Taneva-Popova [46], however, found that while Wang et al.'s approach assumed label independence, contrary to the initial beliefs of leveraging label independencies, joint energy indeed provides the optimal probabilistic approach to address the multi-label OOD problems. Moreover, Wang et al. [39] confirmed that utilizing multiple dominant labels to signal in-distribution inputs effectively captures data features, thus bypassing the need for direct computation and optimization in multi-label datasets. This approach also sidesteps the complexities associated with estimating joint likelihood through generative models, a notably challenging endeavor.

After deriving the *label-wise joint energy* in Eq. 10, we can utilize this method to detect the OOD uncertainty:

$$D(x; \tau) = \begin{cases} \text{out} & \text{if } E_{joint}(x) \leq \tau \\ \text{in} & \text{if } E_{joint}(x) > \tau \end{cases}, \quad (11)$$

where τ is the energy threshold. In our experimental setup, we defined $\tau = 95\%$ to ensure that $D(x; \tau)$ can correctly classify the majority of in-distribution data.

3.3 Spectral Normalized Joint Energy

Based on the foundation laid by Sect. 3.2, we present **Spectral Normalized Joint Energy**(SNoJoE). As part of the feature vector extraction process, spectral normalization is applied to the initial layers of the model. Through power iteration, we evaluate the spectral norm, guaranteeing that the weight matrices of the model adhere to *bi-Lipschitz constraint*.

Firstly, we need to ensure that the spectral norm of the weight matrices $g_l(x) = \sigma(W_l x + b)$ in the non-linear residual blocks of the network is less than 1, thereby ensuring:

$$\|g_l\|_{Lipschitz} \leq \|W_l x + b\|_{Lipschitz} \leq \|W_l\|_2 \leq 1. \quad (12)$$

To achieve this, we apply *spectral normalization* to constrain the weight matrices of the first L layers in the network:

$$W_l = \begin{cases} W_l / \sigma & 1 \leq l \leq L, \\ W_l & l > L \end{cases}, \quad (13)$$

where σ is the spectral norm of the weight matrix, defined as the maximum singular value of the weight matrix. This singular value is obtained through

singular value decomposition (SVD) of the weight matrix. As recommended in [4], *spectral normalization* is used to enforce the weight matrices $\{W_l\}_{l=1}^L$ in Eq. 12 to be *Lipschitz-constrained*, ensuring that the hidden layer parameters $h_i(x)$ “*distance preserving*”.

[3] demonstrates that consider a *hidden mapping* $h : \mathcal{X} \longrightarrow \mathcal{Y}$ with residual architecture $h = h_{L-1} \circ \dots \circ h_2 \circ h_1(x)$ where $h_l(x) = x + g_l(x)$. If for $0 < \alpha \leq 1$, all g_l ’s are α -*Lipschitz*, *i.e.*, $\|g_l(x) - g_l(x')\|_Y \leq \alpha \|x - x'\|_X \quad \forall (x, x') \in \mathcal{X}$. Then:

$$Lips_{lower} * \|x - x'\|_X \leq \|h(x) - h(x')\|_Y \leq Lips_{upper} * \|x - x'\|_X, \quad (14)$$

where $Lips_{lower} = (1 - \alpha)^{L-1}$ and $Lips_{upper} = (1 + \alpha)^{L-1}$ are respectively the lower and upper bounds of *Lipschitz continuity*. Through the *bi-Lipschitz constraint*, the upper bound prevents overfitting during model gradient updates, ensuring the generalization and robustness of the model. The lower bound ensures that there is a certain distance maintained between input feature vectors, *i.e.*, $h(x)$ is *distance preserving*, thereby enabling the extraction of more generalizable features.

Combining the approach from Sect. 3.2, we now update the expression for $h_i(x)$ in Eq. 6:

$$h_i(x) = \begin{cases} \frac{W^{i-1}}{\sigma} \cdot h_{i-1}(x) & 2 \leq i \leq L, \\ W_{i-1} \cdot h_{i-1}(x) & \text{otherwise} \end{cases} \quad . \quad (15)$$

Through the transformation of $h_i(x)$ in Eq. 15, the feature vectors can possess the property of “*distance preserving*” and replace $h_i(x)$ in Eq. 6 to complete the subsequent OOD uncertainty detection.

4 Experiments

In this section, we expound upon our experimental configuration (Sect. 4.1) and showcase the effectiveness of our approach across various out-of-distribution (OOD) evaluation tasks (Sect. 4.2). Furthermore, we delve into ablation studies and conduct comparative analyses, thereby fostering a deeper comprehension of distinct methodologies and ultimately contributing to an enhanced understanding of the field.

4.1 Setup

In-distribution Datasets. We consider the PASCAL-VOC [13] as the in distribution multi-label dataset. It comprises 22,531 images of objects from 20 different categories such as *people*, *dogs*, *cars*, *etc.*, with detailed annotations provided. In this paper, We conduct the OOD detection task to evaluate the performance of our proposed method on this dataset.

Table 1. The dataset configuration in experiments

Dataset	Role	#Classes	#Instances
PASCAL-VOC [13]	In-Distribution (ID)	20	22,531
ImageNet-22K	Out-of-Distribution (OOD)	20 (out of 21841)	18,835
Texture [11]	Out-of-Distribution (OOD)	47	5,640

Training Details. In this study, the multi-label classifier trained is based on the ResNet-101 backbone architecture. The classifier is pretrained on ImageNet-1K [12], and the last layer is replaced by two fully connected layers. Spectral normalization is applied to the first 9 layers of the model. We utilize the Adam optimizer with standard parameters ($\beta_1 = 0.9, \beta_2 = 0.999$), and the initial learning rate during training is set to 1×10^{-4} . Data augmentation techniques such as random cropping and random flipping are employed during training to enhance the dataset, resulting in color images of size 256×256 . After training, the mean Average Precision (mAP) on PASCAL-VOC is 89.19%. The entire experimental process is conducted on NVIDIA GeForce RTX 2080Ti.

Out-of-Distribution Datasets. To evaluate the performance of the model trained on the in-distribution dataset, we employ the Texture dataset [11] and designate 20 classes from ImageNet-22K as out-of-distribution (OOD) datasets. Following the evaluation protocol outlined in [39], we configure the ImageNet-22K dataset in a identical manner for evaluating the PASCAL-VOC pretrained model. The selected classes for evaluation encompass a diverse range, including *dolphin, deer, bat, rhino, raccoon, octopus, giant clam, leech, venus flytrap, cherry tree, Japanese cherry blossoms, redwood, sunflower, croissant, stick cinnamon, cotton, rice, sugar cane, bamboo, turmeric* (Table 1).

Evaluation Metrics. In our experiments, we employ commonly used evaluation metrics for OOD detection under multi-label settings: (i) the false positive rate (FPR95) of OOD examples is calculated when the true positive rate (TPR) of in-distribution examples is held constant at 95%; (ii) the area under the receiver operating characteristic curve (AUROC); (iii) the area under the precision-recall curve (AUPR).

4.2 Results

In Table 2a, we compare our approach with leading OOD detection methods from the literature, showcasing SNoJoE as the new *state-of-the-art* benchmark. Our experimental design carefully selects methods based on pre-trained models to maintain fair comparison standards. Following the guidelines set forth in [39], we evaluated all metrics using the ImageNet dataset for OOD detection.

Additionally, as detailed in Sect. 4.1, we conducted further evaluations using the Texture dataset for OOD detection, with results presented in Table 2b.

Table 2. The comparison of OOD detection performance using spectral normalized joint energy vs. competitive baselines. We use ResNet [17] to train on the in-distribution dataset and use ImageNet-22K (20 classes), Texture [11] as OOD datasets. Besides, \dagger denotes that SNoJoE is statistically better (t-test with p-value < 0.01) than JointEnergy on Texture [11]. All values are percentages. **Bold** numbers are superior results. \uparrow indicates larger values are better, and \downarrow indicates smaller values are better

(a) \mathcal{D}_{out} = ImageNet-22K			
OOD Score	FPR95 \downarrow	AUROC \uparrow	AUPR \uparrow
MaxLogit [7]	36.32	91.04	82.68
MSP [18]	69.85	78.24	67.93
ODIN [24]	36.32	91.04	82.68
Mahalanobis [23]	78.02	70.93	59.84
LOF [6]	76.71	67.54	55.35
Isolation Forest [26]	98.64	41.94	33.50
JointEnergy [39]	31.96	92.32	86.87
SNoJoE(ours)	28.49	93.48	88.11
(b) \mathcal{D}_{out} = Texture [11]			
OOD Score	FPR95 \downarrow	AUROC \uparrow	AUPR \uparrow
MaxLogit [7]	12.36	96.22	96.97
MSP [18]	41.81	89.76	93.00
ODIN [24]	12.36	96.22	96.97
Mahalanobis [23]	19.17	96.23	97.90
LOF [6]	89.49	60.37	76.70
Isolation Forest [26]	99.59	20.89	50.11
JointEnergy [39]	10.87	96.78	97.87
SNoJoE(ours)	5.02\dagger	98.48\dagger	99.00\dagger

Noteworthy, baseline methods like MaxLogit [7], Maximum Softmax Probability (MSP) [18], ODIN [24], and Mahalanobis [23] utilize statistics from the highest values across labels to calculate OOD scores. The Local Outlier Factor (LOF) [6] uses K-nearest neighbors (KNN) to assess local densities, identifying OOD samples through their relatively lower densities compared to neighbors. The Isolation Forest method [26], a tree-based strategy, identifies anomalies by the path lengths from root to terminal nodes. JointEnergy [39] is an energy-based approach that detects OOD instances by evaluating the joint uncertainty among labels.

When conducting OOD detection on different datasets, SNoJoE outperforms several baseline methods across three evaluation metrics. Compared to JointEnergy, which performs OOD detection by utilizing label-wise joint energy, SNoJoE achieves a 11% relative reduction of FPR95 on the subset of ImageNet-22K and a 54% relative reduction of FPR95 on the Texture dataset.

4.3 Ablation Studies

In this section, we delve into a series of ablation experiments to further affirm that neural networks, when subjected to spectral normalization, exhibit a highly regularized feature space. This regularization, in turn, empowers them to identify generalizable features within the data more effectively, thereby enhancing their capability to accurately distinguish out-of-distribution (OOD) data. The observed performance improvement of SNoJoE over JointEnergy, as highlighted in Tables 2a and 2b, underscores that spectral normalization plays a pivotal role in enabling the extraction of more generalizable features from image input space vectors. This enhancement bolsters the model’s proficiency in recognizing OOD samples with greater effectiveness.

In conducting our ablation studies, we persist in utilizing JointEnergy [39] as the benchmark for comparison against our method, SNoJoE. This choice is motivated by the findings presented in Sect. 4.2, where JointEnergy emerged as the most proficient among competing methods, excluding ours. It’s noteworthy that both JointEnergy and SNoJoE capitalize on the joint uncertainty between labels to facilitate OOD detection. For the training configurations and parameters, we adhere to the specifications outlined in Sect. 4.1.

Additionally, we explored the application of spectral normalization across various layers of the network structure to gauge its influence on multi-label OOD detection tasks. Our experimental findings, detailed in Tables 3a and 3b, involved implementing spectral normalization at different levels within the ResNet framework [17] to assess its effect on OOD detection. The results suggest that indiscriminate use of spectral normalization could, in some cases, impair the model’s ability to perform multi-label OOD detection effectively. Specifically, when spectral normalization is limited to the first seven layers of the network (refer to the second row of Tables 3a and 3b), the model’s efficacy may decline compared to a non-normalized version. This deterioration in performance might stem from the application of spectral normalization solely to the network’s more superficial layers. Given that these initial layers process simpler representations, imposing stringent constraints on them could diminish the network’s capacity for expressive representation, thereby undermining its performance. Conversely, extending spectral normalization to the model’s deeper layers (as illustrated in the last two rows of Tables 3a and 3b) appears to enhance the model’s proficiency in learning and capturing intricate input vector features. This improvement is likely due to the advanced abstraction abilities of the deeper layers. However, this finding should not be misconstrued to suggest that greater application of spectral normalization invariably results in superior performance, as it also increases computational demands. Furthermore, the practicality of applying spectral normalization to certain layers (such as those involved in average pooling to decrease spatial dimensions of feature maps) remains questionable, given the negligible benefits it may offer.

In summary, our experiments reveal that indiscriminate use of spectral normalization across the network does not invariably enhance the model’s performance and might even impair it. Nevertheless, if spectral normalization is judi-

Table 3. Ablation study on the impact of the numbers of layers applied spectral normalization using ImageNet-22K (20 classes) and Texture [11] as OOD datasets. #layers are numbers of layers applied spectral normalization

(a) \mathcal{D}_{out} : ImageNet-22K (20 classes)			
#layers	FPR95↓	AUROC↑	AUPR↑
0	31.37	93.37	89.29
7	48.19	89.59	84.19
8	30.85	92.98	87.24
9	28.49	93.48	88.11

(b) \mathcal{D}_{out} : Texture [11]			
#layers	FPR95↓	AUROC↑	AUPR↑
0	6.21	97.87	98.58
7	6.72	98.20	98.94
8	4.91	98.49	98.94
9	5.02	98.48	99.00

ciuously applied to enable the network to more effectively learn complex and generalizable features from the input vectors, the model’s performance surpasses that of models without spectral normalization. The performance discrepancy can reach as high as 2.88 and 1.30 in FPR95, with the OOD dataset being ImageNet-22K and Texture, respectively.

5 Conclusion

In this study, we introduce a cutting-edge method for OOD detection named Spectral Normalized Joint Energy (SNoJoE) in the context of multi-label classification.

Our findings reveal that spectral normalization applied to the initial layers of a pre-trained model’s network significantly enhances model robustness, improves generalization capabilities, and more effectively distinguishes between in-distribution and out-of-distribution inputs. When compared to leading baseline approaches, SNoJoE sets a new benchmark for OOD detection, establishing itself as the new *state of the art* in this domain, while not substantially increasing computational demands.

We anticipate that our contribution will spark further exploration into multi-label OOD detection and encourage the expansion of this research area into wider applications.

Acknowledgement. This work was supported by National Key R&D Program of China (No. 2021YFC3340700), NSFC grant (No. 62136002), Ministry of Education Research Joint Fund Project (8091B042239), Shanghai Knowledge Service Platform Project (No. ZF1213), and Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. *Cogn. Sci.* **9**(1), 147–169 (1985)
2. Ayadi, A., Ghorbel, O., Obeid, A.M., Abid, M.: Outlier detection approaches for wireless sensor networks: a survey. *Comput. Netw.* **129**, 319–333 (2017)
3. Bartlett, P.L., Evans, S.N., Long, P.M.: Representing smooth functions as compositions of near-identity functions with implications for deep network optimization (2018)
4. Behrmann, J., Grathwohl, W., Chen, R.T.Q., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks (2019)
5. Bendale, A., Boult, T.: Towards open set deep networks (2015)
6. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. *SIGMOD Rec.* **29**(2), 93–104 (2000). <https://doi.org/10.1145/335191.335388>
7. Chan, R., et al.: Segmentmeifyoucan: a benchmark for anomaly segmentation (2021)
8. Chaudhari, S., Shevade, S.: Learning from positive and unlabelled examples using maximum margin clustering. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) *ICONIP 2012. LNCS*, vol. 7665, pp. 465–473. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34487-9_56
9. Chen, L.C., Schwing, A.G., Yuille, A.L., Urtasun, R.: Learning deep structured models (2015)
10. Chen, L., Zhan, W., Tian, W., He, Y., Zou, Q.: Deep integration: a multi-label architecture for road scene recognition. *IEEE Trans. Image Process.* **28**(10), 4883–4898 (2019). <https://doi.org/10.1109/TIP.2019.2913079>
11. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild (2013)
12. 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)
13. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vision* **111**, 98–136 (2015)
14. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 195–200 (2005)
15. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation (2014)
16. He, F., Liu, T., Webb, G.I., Tao, D.: Instance-dependent PU learning by Bayesian optimal relabeling (2020)
17. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks (2016)
18. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks (2018)
19. Hinz, T., Heinrich, S., Wermter, S.: Generating multiple objects at spatially distinct locations (2019)
20. Hsu, Y.C., Shen, Y., Jin, H., Kira, Z.: Generalized ODIN: detecting out-of-distribution image without learning from out-of-distribution data (2020)

21. Huang, R., Geng, A., Li, Y.: On the importance of gradients for detecting distributional shifts in the wild (2021)
22. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. *Predicting Structured Data* **1** (2006)
23. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks (2018)
24. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks (2020)
25. Lin, Z., Roy, S.D., Li, Y.: Mood: multi-level out-of-distribution detection (2021)
26. 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>
27. Liu, S.M., Chen, J.H.: A multi-label classification based approach for sentiment classification. *Expert Syst. Appl.* **42**(3), 1083–1093 (2015)
28. Liu, W., Wang, X., Owens, J.D., Li, Y.: Energy-based out-of-distribution detection (2021)
29. Menon, A., Rooyen, B.V., Ong, C.S., Williamson, B.: Learning from corrupted binary labels via class-probability estimation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, Lille, France, vol. 37, pp. 125–134. PMLR (2015). <https://proceedings.mlr.press/v37/menon15.html>
30. Morteza, P., Li, Y.: Provable guarantees for understanding out-of-distribution detection (2021)
31. Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., Samatova, N.F.: Anomaly detection in dynamic networks: a survey. *Wiley Interdisc. Rev. Comput. Stat.* **7**(3), 223–247 (2015)
32. Ranzato, M., Poulnary, C., Chopra, S., Cun, Y.: Efficient learning of sparse representations with an energy-based model. In: Advances in Neural Information Processing Systems, vol. 19 (2006)
33. Reiss, T., Cohen, N., Bergman, L., Hoshen, Y.: Panda: adapting pretrained features for anomaly detection and segmentation (2021)
34. Ruff, L., et al.: Deep one-class classification. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4393–4402. PMLR (2018). <https://proceedings.mlr.press/v80/ruff18a.html>
35. Scott, C.: A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In: Lebanon, G., Vishwanathan, S.V.N. (eds.) Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, San Diego, California, USA, vol. 38, pp. 838–846. PMLR (2015). <https://proceedings.mlr.press/v38/scott15.html>
36. Tax, D.M.J.: One-class classification: concept learning in the absence of counter-examples. (2002)
37. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. *Int. J. Data Warehous. Mining (IJDWM)* **3**(3), 1–13 (2007)
38. Tu, L., Gimpel, K.: Learning approximate inference networks for structured prediction (2018)
39. Wang, H., Liu, W., Bocchieri, A., Li, Y.: Can multi-label classification networks know what they don't know? (2021)
40. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: a unified framework for multi-label image classification (2016)

41. Wang, J., Cherian, A.: Gods: generalized one-class discriminative subspaces for anomaly detection (2019)
42. Wei, H., Xie, R., Cheng, H., Feng, L., An, B., Li, Y.: Mitigating neural network overconfidence with logit normalization (2022)
43. Wei, Y., et al.: HCP: a flexible CNN framework for multi-label image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(9), 1901–1907 (2016). <https://doi.org/10.1109/TPAMI.2015.2491929>
44. Xie, J., Lu, Y., Zhu, S.C., Wu, Y.N.: A theory of generative convnet (2016)
45. Yang, X., Latecki, L.J., Pokrajac, D.: Outlier detection with globally optimal exemplar-based GMM. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 145–154. SIAM (2009)
46. Zhang, D., Taneva-Popova, B.: A theoretical analysis of out-of-distribution detection in multi-label classification. In: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 275–282 (2023)
47. Zhang, H., Li, A., Guo, J., Guo, Y.: Hybrid models for open set recognition (2020)
48. Zhang, W., Yan, J., Wang, X., Zha, H.: Deep extreme multi-label learning (2018)
49. Zhong, J.X., Li, N., Kong, W., Liu, S., Li, T.H., Li, G.: Graph convolutional label noise cleaner: train a plug-and-play action classifier for anomaly detection (2019)



Noisy Label Learning Based on Weighted Neighborhood Consistency

Qian Rong¹, Lu Zhang², Ling Yuan^{1(✉)}, Xuanang Ding¹, and Guohui Li³

¹ School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

{qianrong, cherryyuanling, dingxuanang}@hust.edu.cn

² School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

luzhang_cs@hust.edu.cn

³ School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China

guohuili@hust.edu.cn

Abstract. In the realm of deep learning applied to real-world scenarios, the existence of noisy labels is an inevitable factor that can detrimentally affect the models' performance. Most state-of-the-art methods for learning from noisy labels rely on sample selection strategies that partition the training data into clean and noisy labeled samples. Subsequently, these noisy label samples are treated as unlabeled samples, and the empirical vicinal risk is minimized through semi-supervised learning. Therefore, accurately identifying noisy labels contributes to enhancing the performance of the model. To enhance the accuracy and stability of sample selection, this paper proposes utilizing the mean and variance of the loss sequence to identify clean samples and noisy ones. Nonetheless, sample selection is not entirely effective in eliminating noisy label samples, as a small fraction of them are inadvertently considered as clean samples. Consequently, we propose Weighted Neighborhood Consistency Regularization (WNCR), which alleviates the impact of residual noisy labels by encouraging the neural network to maintain consistency in its predictions with those of its k -nearest neighbors for each sample. Extensive experiments on multiple synthetic and real-world noisy datasets demonstrate that our method outperforms the state-of-the-art methods at multiple noise levels.

Keywords: Neighborhood Consistency · Label noise · Sample selection · semi-supervised learning

Supported by the National Natural Science Foundation of China under Grant No. 62272180 and No. 62272176. The computation is completed in the HPC Platfrom of Huazhong University of Science and Technology.

1 Introduction

Deep neural networks have shown remarkable success in computer vision tasks [20], information retrieval [25], and natural language processing [16]. These successes are dependent on the large amounts of accurately annotated training data. However, obtaining a large amount of accurately annotated training data in real application scenarios is challenging due to its high cost and time-consuming. Online querying [4] and crowdsourcing [27] are frequently employed as substitutes for manual annotation in training data annotation. Nevertheless, these methods often result in the introduction of label noise. Deep neural networks possess strong data fitting capabilities, making them prone to overfitting noisy label data, consequently resulting in a decline in their generalization performance [40]. Hence, it is imperative to conduct investigations and develop robust algorithms that can effectively handle learning from noisy labels.

The current strategies employed to address label noise include the use of robust loss functions [10, 34], noise transition matrix [11, 23], and sample selection [7, 31]. The sample selection-based noise label learning method aims to distinguish between noisy and clean data, subsequently handling them differently. The effectiveness of sample selection-based methods relies on accurately distinguishing between noisy labeled samples and clean samples. Previous studies have frequently employed the principle of small loss to differentiate between noisy labeled samples and clean samples. Specifically, in the training process, samples exhibiting small loss values are identified as clean samples, while samples with large loss values are identified as noisy labeled samples. However, given that neural networks are typically optimized using stochastic gradient descent, the training process inherently possesses randomness, which leads to fluctuations in the loss values of training samples. Specifically, this can manifest in clean samples experiencing sudden spikes in loss during training, while noisy samples might see their loss values plummet unexpectedly, which does not adhere to the principle of small loss. Hence, there is a random error in distinguishing clean samples from noisy ones based on the sample loss after a single neural network update. To mitigate this random error, we identify clean samples and noisy ones by examining the mean and variance of the entire loss sequence. Current sample selection methods are insufficient to completely and thoroughly separate clean samples from noisy ones, resulting in a small fraction of noisy labels still present among the supposedly clean training samples. Consequently, we propose a Weighted Neighborhood Consistency Regularization (WNCR) to suppress the impact of residual noisy labels. The WNCR encourages the neural network to maintain consistency in its predictions for training samples with their true k -nearest neighbors.

In this paper, we utilize the co-training strategy to simultaneously train two neural networks with identical structures. Initially, both networks undergo a warm-up process. Subsequently, the sample selection is performed using the current loss sequence, after the completion of each epoch of training. Lastly, the two networks exchange the identified clean and noisy labeled samples and proceed with semi-supervised learning individually.

Our main contributions are the following:

- 1) We propose sample selection based on the mean and variance of the loss sequence, which can mitigate random errors, thereby enhancing the accuracy and stability of sample identification.
- 2) We propose Weighted Neighborhood Consistency Regularization (WNCR) to alleviate the detrimental effects of noisy labels, by encouraging the neural network to maintain consistency in its predictions with the true k-nearest neighbors of the samples.
- 3) We conduct experiments on various synthetic (CIFAR10 and CIFAR100) and real noise labels (CIFAR10N, CIFAR100N and Clothing1M) to validate the rationality and effectiveness of our proposed method.

2 Related Works

In this section, we review previous research from the perspective of solving the noisy label problem. We group recent research efforts into noise modeling, sample selection, and loss design.

Noise Modeling. This type of work considers that noise labels obey specific rules, and hopes to simulate this rule through parametric models. The main direction is to estimate the underlying noise transition matrix [6, 12, 15]. Evaluating such a noisy transition matrix and modeling noisy labels typically necessitates either additional clean data or specific assumptions [5, 36, 41].

Sample Selection. The most intuitive way to solve the noise label is to identify the noise label so that the harm caused by the noise label can be avoided. However, it is not easy to identify noise labels. Literature [2] found that the neural network learns clean samples first, and then learns noise labels. Therefore, many research works regard small loss samples as clean samples and large loss samples as noisy label samples, named the small loss principle. Some research work proposes to train two models at the same time, and then during the training process, the two models respectively use the sample recognition method designed based on the principle of small loss to input the recognized clean samples into the other model [13, 18, 29, 33]. However, these methods do not make full use of noisy label samples. Later studies proposed to perform sample identification first, treat noisy labeled samples as unlabeled data, and then perform semi-supervised learning, where Semi-Supervised Learning (SSL) based methods tend to show competitive results on benchmarks, such as [1, 8, 21]. In addition, some studies choose to perform self-supervised learning first, then perform sample identification, and finally perform semi-supervised fine-tuning [7, 26, 28]. The performance of the above methods largely depends on the accuracy of sample identification and the performance of subsequent semi-supervised learning.

Loss Design. We consider both designing robust loss functions [10, 32] and regularization constraints [17, 30] as loss designs. For example, [24] proposed a new

family of loss functions called peer loss, where performing empirical risk minimization on a corrupted dataset is equivalent to performing empirical risk minimization on the corresponding clean dataset. [17] proposes a sample neighbor consistency regularization that leverages similarities between training examples in feature space, encouraging the prediction of each example to be similar to its nearest neighbors. However, the method of loss design can alleviate the noisy label problem, but it cannot really solve the noisy label problem.

3 Preliminary

3.1 Problem Formulation

For a multi-class classification task with c classes, we consider a dataset $D = \{(x_i, \hat{y}_i)\}_{i=1}^n$ with n samples, where x_i is the sample, e.g. an image, has a corresponding annotation label $\hat{y}_i \in \mathcal{Y} = \{1, 2, \dots, c\}$. In practice, \hat{y}_i might be a noise label, which is not equal to the ground-truth label y_i of x_i . Additionally, we don't know whether the annotation label \hat{y}_i is noisy ($\hat{y}_i \neq y_i$) or clean ($\hat{y}_i = y_i$) during training. The goal of the multi-class classification task is to learn a model M that takes an example x_i as the input and outputs the ground-truth label of x_i . In a mathematical formulation, the objective of a classification task can be described as follows:

$$\min_{\theta, \phi} \frac{1}{|D|} \sum_{i=1}^{|D|} \ell(\tau(f(g(x_i; \phi); \theta)), y_i), (x_i, y_i) \in D \quad (1)$$

The classification model $M = f_\theta \circ g_\phi$ consists of a feature extractor g_ϕ and a classification head f_θ , where ϕ and θ represent the parameters of g_ϕ and f_θ respectively. For convenience, f and g are denoted by f_θ and g_ϕ in the sequel. Where ℓ represents the loss function, wherein the widely adopted choice is the cross-entropy loss function, y_i signifies the ground-truth of x_i , τ is the softmax function.

4 Proposed Method

Our proposed method contains two main parts, sample selection and Weight Neighborhood Consistency Regularization. The sample selection process involves dividing the training samples into clean and noisy subsets, subsequently treating the clean samples as labeled and the noisy ones as unlabeled for semi-supervised training purposes. Sample selection cannot entirely eliminate noisy label samples, so we further propose Weighted Neighborhood Consistency Regularization to mitigate the harm caused by residual noisy label samples.

4.1 Sample Selection

Currently, the most commonly used sample selection method is the small-loss criterion, which regards samples with low loss values as clean samples and those with high loss values as noisy samples. However, most neural networks are optimized using stochastic gradient descent, which introduces randomness into the parameter updates, further making the loss values of training samples inherently stochastic. In specific, the loss value of training samples is fluctuating. During training, the loss value of clean samples can sometimes be large, while the loss value of noisy samples can also be small. Consequently, this paper proposes to select samples based on the loss sequence, which can help mitigate the randomness in loss values.

After each epoch of training, we compute the loss for the entire training dataset, forming a sequence set of losses, which we denote as $L = \{l_0, l_1, \dots, l_n\}$, where $l_i = \{l_{i0}, l_{i1}, \dots, l_{it}\}$ is the sequence of losses for sample x_i , t represents the end of training for the t th epoch. For each training example x_i , we compute the mean m_i and variance v_i of the loss sequence l_i as follows:

$$m_i = \frac{1}{|l_i|} \sum_{j=1}^{|l_i|} l_{ij} \quad (2)$$

$$v_i = \frac{1}{|l_i|} \sum_{j=1}^{|l_i|} (l_{ij} - m_i)^2 \quad (3)$$

We combine m_i and v_i to form a 2-dimensional vector s_i to distinguish whether a training sample x_i is a clean sample or not, as follows:

$$s_i = \text{Normalize}(m_i, v_i) \quad (4)$$

Normalize stands for data normalization. Each training sample x_i corresponds to a two-dimensional vector s_i , and all the vectors s_i form a set S , as follows:

$$S = \{s_0, s_1, \dots, s_n\} \quad (5)$$

We apply the K-means clustering algorithm to cluster the set S into two subsets, S_0 and S_1 . All training samples corresponding to elements in set S_0 compose the training subset D_0 , and all training samples corresponding to elements in set S_1 form the training subset D_1 . For any $s_i \in S_0$, there exists $x_i \in D_0$ and for any $s_i \in S_1$, there exists $x_i \in D_1$. We identify the sets of clean samples D_c and noise samples D_n as follows:

$$D_c = \begin{cases} D_0 & \text{if } \frac{1}{|D_0|} \sum \|s_i\|_2 < \frac{1}{|D_1|} \|s_j\|_2, s_i \in S_0, s_j \in S_1 \\ D_1 & \text{if } \frac{1}{|D_0|} \sum \|s_i\|_2 > \frac{1}{|D_1|} \|s_j\|_2, s_i \in S_0, s_j \in S_1 \end{cases} \quad (6)$$

$$D_n = \begin{cases} D_0 & \text{if } \frac{1}{|D_0|} \sum \|s_i\|_2 > \frac{1}{|D_1|} \|s_j\|_2, s_i \in S_0, s_j \in S_1 \\ D_1 & \text{if } \frac{1}{|D_0|} \sum \|s_i\|_2 < \frac{1}{|D_1|} \|s_j\|_2, s_i \in S_0, s_j \in S_1 \end{cases} \quad (7)$$

where $\|\cdot\|$ represents the L2 norm.

4.2 Weighted Neighborhood Consistency Regularization

After dividing the training dataset into clean training data D_c and noisy training data D_n , we treat the clean training data as labeled and the noisy training data as unlabeled, and then proceed with semi-supervised learning. Semi-supervised learning can mitigate the impact of noisy labels, but existing sample selection methods are not perfect at differentiating clean samples from noisy ones. Consequently, there are still a few noisy samples within the clean training data set D_c . Inspired by [17], we propose Weighted Neighborhood Consistency Regularization to further mitigate the impact of residual noisy samples within the clean data set D_c .

A good neural network should be stable and smooth, meaning that it produces similar outputs for similar inputs. As a result, [17] proposes a method called Neighbor Consistency Regularization (NCR) to find the k-nearest neighbors for each sample in feature space, and constrains the neural network to output for the sample consistent with the outputs of its k-nearest neighbors, as fellow:

$$\mathcal{L}_{NCR} = \frac{1}{|D|} \sum_{i=0}^{|D|} KL(\tau(o_i/T) \parallel \sum_{j \in N_k(x_i)} \frac{d_{i,j}}{\sum_k d_{i,k}} \cdot \tau(o_j/T)) \quad (8)$$

$$d_{i,j} = \cos(z_i, z_j) \quad (9)$$

$$z_i = g(x_i), z_j = g(x_j), o_i = f(z_i), o_j = f(z_j) \quad (10)$$

where KL is the KL-divergence to measure the difference between two distributions, $N_k(x_i)$ is the k-nearest neighbor samples of sample x_i in the feature space, T is the hyperparameter temperature, τ is is the softmax function.

In practical applications, the feature space of samples typically has a high dimensionality. However, distance measures degrade as the dimensionality increases in high-dimensional spaces, making it inadequate to precisely determine the proximity between samples [9]. Consequently, the k-nearest neighbors found by NCR in the feature space using cosine distance might be unreliable (not necessarily true neighbors) and noisy labels can further degrade the reliability of these k-nearest neighbors. If a sample's k-nearest neighbors are unreliable, enforcing similarity between the network's prediction for the sample and those of its unreliable neighbors does not necessarily enhance the network's consistency and performance. Therefore, we propose a Weighted Neighborhood Consistency Regularization (WNCR) based on NCR, as fellow:

Algorithm 1. Training Workflow

Input: training set D , Neural network M_1 , Neural network M_2 , Maximum training epoch E

- 1: warm up M_1 and M_2
- 2: **if** $epoch < E$ **then**
- 3: D_c^1 and $D_n^1 \leftarrow$ sample selection with M_1
- 4: D_c^2 and $D_n^2 \leftarrow$ sample selection with M_2
- 5: M_1 Train with the Equation (20) on D_c^1 and D_n^1
- 6: M_2 Train with the Equation (20) on D_c^1 and D_n^1

Output: Neural network \hat{M}_1 , Neural network \hat{M}_2

$$\mathcal{L}_{WNCR} = \frac{1}{\sum_{i=0}^{|D|} w_i} \sum_{i=0}^{|D|} w_i \cdot KL(\tau(o_i/T) || \sum_{j \in N_k(x_i)} \frac{d_{i,j}}{\sum_k d_{i,k}} \cdot \tau(o_j/T)) \quad (11)$$

$$\omega_i = \begin{cases} 1 & \text{if } entroy(\tilde{y}_i) < \sigma \\ 0 & \text{else} \end{cases} \quad (12)$$

$$\tilde{y}_i = (\tilde{y}_{i0}, \tilde{y}_{i1}, \dots, \tilde{y}_{ic}) \quad (13)$$

$$\tilde{y}_{ie} = \frac{Num(y_j = k)}{|N_k^c(x_i)|}, \quad x_j \in N_k^c(x_i) \quad (14)$$

$$N_k^c(x_i) = N_k(x_i) \cap D_c \quad (15)$$

where w_i represents the weight coefficient, which takes a value of either 0 or 1. \tilde{y}_{ie} denotes the proportion of k-nearest neighbors of sample x_i with label e among all k-neighbors. Our proposed method WNCR enforces neighborhood consistency only on training samples with reliable k-nearest neighbors. Reasonably, a reliable k-neighbor of a sample is likely to have the same ground-truth label as the sample itself. Hence, we posit that the more consistent the labels of a sample's k-nearest neighbors, the more reliable they are, and conversely. We employ information entropy to measure the degree of label consistency among a sample's k-nearest neighbors. If $entroy(\tilde{y}_i) < \sigma$, then we consider the nearest neighbor $N_k(x_i)$ of sample x_i to be reliable, σ is a threshold. The smaller $entroy(\tilde{y}_i)$ is, the more consistent the labels of sample x_i 's k-nearest neighbors tend to be.

4.3 Training and Objective Functions

We consider D_c and D_n as the labeled and unlabeled datasets, respectively. We then perform semi-supervised learning Mixmatch [3]. The objective function of semi-supervised learning is as follows:

$$\mathcal{L}_f = \mathcal{L}_{X_i} + \frac{1}{\lambda} \mathcal{L}_{U_i} \quad (16)$$

$$\mathcal{L}_{X_i} = \frac{1}{|X'|} \sum_{x,p \in X'} CE(p, f(g(x))) \quad (17)$$

$$\mathcal{L}_{U_i} = \frac{1}{L|U'|} \sum_{u,q \in U'} \|q - f(g(u))\|_2^2 \quad (18)$$

$$X', U' = MixMatch(D_k^c, D_k^n, T, D, \beta) \quad (19)$$

CE denotes cross-entropy and λ_U , T , D , β are the hyperparameters, which are associated with semi-supervised learning, exhibit minimal correlation with the process of learning from noisy labels. As a result, this paper will not specifically analyze these hyperparameters.

The final objective to minimized during the training:

$$\mathcal{L} = (1 - \alpha) \mathcal{L}_f + \alpha \mathcal{L}_{WNCR} \quad (20)$$

This paper employs co-training to simultaneously train two initially distinct but structurally identical neural networks, both optimizing the same objective function \mathcal{L} . First, the neural networks M_1 and M_2 are warm-up. At the end of each epoch, M_1 partitions the training data into clean set D_c^1 and noisy set D_n^1 through sample selection, while M_2 separates the data into clean set D_c^2 and noisy set D_n^2 using the same approach. Feed the clean samples set D_c^1 and noisy samples set D_n^1 from Neural Network M_1 to Neural Network M_2 for semi-supervised learning, and vice versa, As shown in Algorithm 1.

5 Experiments

5.1 Experimental Setup

Dataset. Our experiments were conducted on the CIFAR10 [19], CIFAR100 [19], CIFAR10N [35], CIFAR100N [35], and Clothing1M [37] datasets. Data sets CIFAR10 and CIFAR10N, as well as CIFAR100 and CIFAR100N, are distinct from each other. The CIFAR10N and CIFAR100N datasets are real human-annotated noise labels collected from Amazon Mechanical Turk by Wei et al. [35]. CIFAR10N contains five versions with different noise levels, and CIFAR100N contains two versions with different levels. Clothing1M contains 1M 224×224 clothing images in 14 classes. It is a dataset with noisy labels since the data is collected from several online shopping websites and include many mislabelled samples.

Implementation Details. We use the ResNet18 [14] network for CIFAR10 and CIFAR10N and the ResNet34 [14] network for CIFAR100 and CIFAR100N. For Clothing1M, we use a ResNet50 [14] network pre-trained on ImageNet. These networks are trained for 600 epochs, with a warm-up period of 10 epochs for

CIFAR-10 and 30 epochs for CIFAR-100. We employ SGD as the optimizer with a momentum of 0.9 and weight decay of 0.001. The batch size is fixed as 256 and the initial learning rate is 0.05, which decays by a cosine scheduler. We set, $k = 20$, $\sigma = 1.5$ for CIFAR10 and CIFAR10N, $\sigma = 3$ for CIFAR100 and CIFAR100N, $\alpha = 0.2$, $T = 2$.

Baseline. We compare our method with multiple state-of-the-art methods. Specifically, they are Co-Teaching+ [39], M-correction [1], PENCIL [38], JoCoR [33], DivideMix [21], SOP+ [22], NCR [17] and LongReMix [8].

Table 1. Best accuracy (%) comparisons on CIFAR10 and CIFAR100 with symmetric and asymmetric noise. We report the best accuracy of our method.

Dataset	CIFAR10					CIFAR100				
Noise type	symmetry				asymmetry	symmetry				
Methods\Noise Ratio	0.2	0.5	0.8	0.9	0.4		0.2	0.4	0.8	0.9
CE	86.8	79.4	62.9	42.7	85.0		62.0	46.7	19.9	10.1
Co-Teaching+	89.5	85.7	67.4	47.9	-		65.6	51.8	27.9	13.7
DivideMix	96.1	94.6	93.2	76.0	93.4		77.3	74.6	60.2	31.5
JoCoR	85.7	79.4	27.8	-	76.4		53.0	43.5	15.5	-
M-correction	94.0	92.0	86.8	69.1	87.4		73.9	66.1	48.2	24.3
PENCIL	92.4	89.1	77.5	58.9	88.5		69.4	57.5	31.1	15.3
LongReMix	96.2	95.0	93.9	82.0	94.7		77.8	75.6	62.9	33.8
SOP+	96.3	95.5	94.0	-	93.8		78.8	75.9	63.3	-
NCR	95.2	94.3	91.6	75.1	90.7		76.6	72.5	58.0	30.8
ours	96.85	96.2	94.54	89.21	95.24		79.62	76.51	64.85	35.76

5.2 Comparison with State-of the-Art Methods

We compare with multiple state-of the-art noisy label learning methods in both synthetic and natural label noise Settings. For their performance, we directly adopt the reported results from their receptive papers.

We synthetically generate noisy labels on the CIFAR10 and CIFAR100 datasets, considering both symmetric and asymmetric noise types, as well as multiple noise levels. As shown in Table 1, our method consistently exhibits the best classification performance across different settings, substantiating the superiority of our proposed approach.

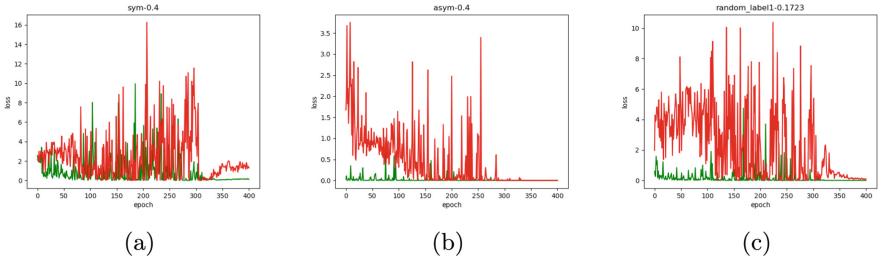
CIFAR10N, CIFAR100N, and Clothing1M are three real-world noisy label datasets, where the noisy labels stem from human annotation errors. As shown in Table 2 and Table 3, Our proposed method consistently outperforms baseline approaches in terms of classification performance when dealing with real-world noisy labels, demonstrating its superiority and advancement.

Table 2. Accuracy comparisons on CIFAR-10N and CIFAR-100N under different noise types.

Dataset	CIFAR10N			CIFAR100N	
Methods	Aggre	Rand1	Worst	Noisy	Fine
CE	87.77	85.02	77.69	55.50	
Co-Teaching+	91.20	90.33	83.83	60.37	
DivideMix	95.01	95.16	92.56	71.13	
JoCoR	91.44	90.30	83.37	59.97	
SOP+	95.61	95.28	93.24	67.81	
ours	96.35	96.12	94.14	72.21	

Table 3. Accuracy (%) comparisons on Clothing1M dataset.

Methods	CE	DivideMix	JoCoR	PENCIL	LongReMix	SOP+	NCR	ours
Accuracy	69.21	74.76		70.03	73.49	74.38	73.50	74.6 74.82

**Fig. 1.** A sample loss chart, with the red line representing noisy samples and the green line representing clean samples. (Color figure online)

5.3 The Effectiveness of Sample Selection

In this subsection, we validate the rationality and effectiveness of the sample selection method proposed in this paper through experiments from multiple aspects. As shown in Fig. 1, we plot the loss values of a randomly selected clean sample and a noisy sample individually, where drawing more randomly selected samples would make the chart too cluttered to be readable. From Fig. 1, we can observe that both the loss of clean samples and that of noisy samples are fluctuating. At certain moments, the loss of clean samples might spike, while the loss of noisy samples could drop significantly.

Upon observation, we notice that the loss values of clean samples are generally smaller and less volatile compared to those of noisy samples. Consequently, we contemplate using the mean and variance of the loss sequences for sample selection. From Fig. 2, we can observe that the mean and variance of the clean sample loss sequences exhibit statistically significant differences compared to

those of the noisy samples. Therefore, it is reasonable to employ the mean and variance of the loss sequences for sample selection.

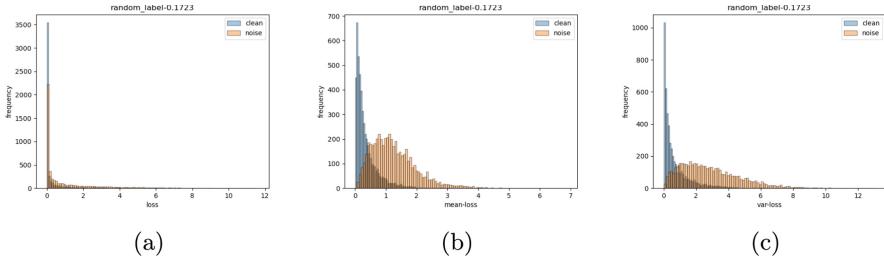


Fig. 2. (a) displays a histogram of the loss values for clean and noisy samples, while (b) represents a histogram of the mean values of the loss sequences, and (c) shows a histogram of the variances of the loss sequences.

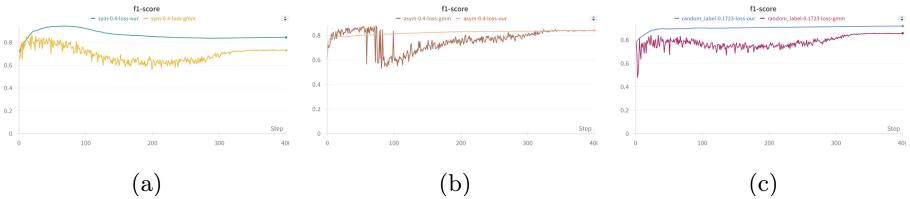


Fig. 3. F1 scores for identifying clean samples and noisy samples using the sample selection method under different noise settings.

In Fig. 3, “loss-gmm” represents the sample selection method commonly employed in many papers [8], which clusters loss values using a Gaussian distribution model, “loss-our” representing the sample selection method proposed in this paper. It can be observed that our proposed fundamental selection method generally outperforms and exhibits greater stability in most cases. In conclusion, the sample selection method we propose is both rational and effective.

5.4 The Effectiveness of WNCR

In this section, we demonstrate the limitations of NCR through experiments and validate the rationality and effectiveness of our proposed WNCR. Figure 4 illustrates the consistency of labels for k -nearest neighbors found using cosine distance, where lower entropy indicates higher consistency in labels among the k -nearest neighbors, and higher entropy indicates greater label inconsistency. It is evident that there are always some samples whose k -nearest neighbors have inconsistent labels, and these neighbors are likely not genuine k -nearest neighbors for those samples. Therefore, it is reasonable for our proposed WNCR to only perform neighborhood consistency regularization on reliable k -nearest neighbors.

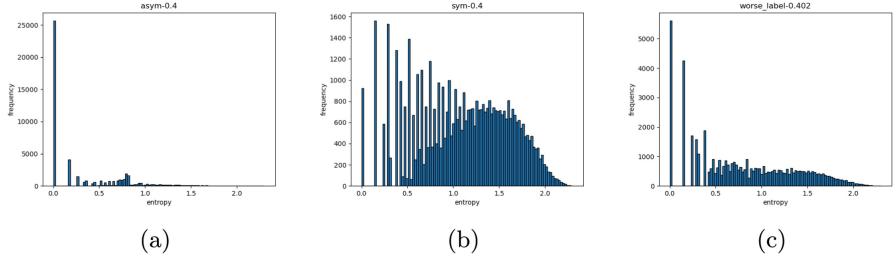


Fig. 4. The consistency level of k-nearest neighbors found using cosine distance under different noise settings.

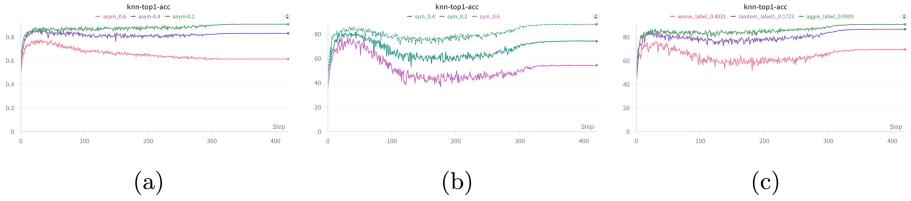


Fig. 5. Top-1 accuracy of the KNN algorithm under different noise settings.

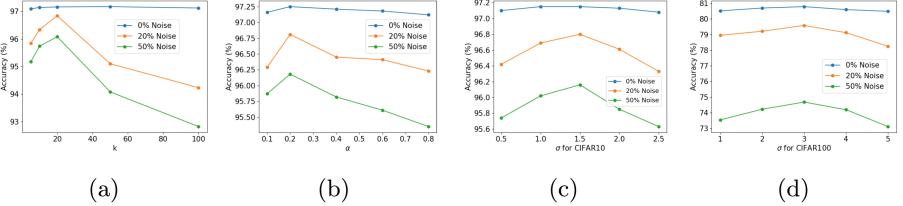


Fig. 6. Impact of hyperparameters, k , α , and σ are evaluated on the CIFAR-10 validation set and the impact of hyperparameters σ on the CIFAR-10 validation set.

As clearly observed in Fig. 5, noise labels can decrease the top-1 accuracy of k-nearest neighbors, leading to some samples in feature space having k-nearest neighbors that are not genuine neighbors. Thus, our proposed WNCR approach is justified.

In Table 4, “Standard” means that samples are selected using the most basic small loss principle and then semi-supervised training is performed. In this paper, the sample selection method proposed is represented by “SR”, the semi-supervised learning algorithm MixMatch is denoted by “Mix”. “NCR” and “WNCR” denote the s Neighbor Consistency Regularization(NCR) [17] and the Weighted Neighborhood Consistency Regularization(WNCR), respectively. As shown in Table 4, the effectiveness of WNCR can be observed.

Table 4. The impact of important components

Dataset	CIFAR10		CIFAR100		CIFAR10N		CIFAR100N
noise type	sym	sym	sym	sym	Aggre	Worst	Fine
noise ratio	20%	50%	20%	40%	9.03%	40.21%	40.2%
Standard	88.11	77.34	61.65	38.92	91.18	77.53	38.41
SR+Mix	93.63	92.16	74.34	72.55	92.43	90.88	68.48
SR+Mix+NCR	96.34	95.42	78.85	75.16	95.87	93.51	71.15
SR+Mix+WNCR	96.85	96.2	79.62	76.51	96.35	94.14	72.21

5.5 Hyperparameter Analysis

Figure 6 illustrates the validation accuracy at various noise rates for different values of the significant hyperparameters. We can see that our hyperparameters are not particularly sensitive to the impact of performance, and no hyperparameter changes lead to large changes in validation accuracy.

6 Conclusion

This paper aims to tackle the issue of noisy label learning from two perspectives: enhancing the accuracy of recognizing noisy labeled samples and augmenting the robustness of neural networks. We propose a method for sample selection based on loss sequences to identify noisy label samples. Sample selection alone is not sufficient to completely eliminate noisy label samples, so we introduce a weighted neighborhood consistency regularization to mitigate the impact of residual noisy label samples.

References

1. Arazo, E., Ortego, D., Albert, P., O’Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: International Conference on Machine Learning, pp. 312–321. PMLR (2019)
2. Arpit, D., et al.: A closer look at memorization in deep networks. In: International Conference on Machine Learning, pp. 233–242. PMLR (2017)
3. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: a holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
4. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM (JACM) **50**(4), 506–519 (2003)
5. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1431–1439 (2015)
6. Cheng, D., et al.: Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16630–16639 (2022)

7. Cordeiro, F.R., Belagiannis, V., Reid, I., Carneiro, G.: Propmix: hard sample filtering and proportional mixup for learning with noisy labels. arXiv preprint [arXiv:2110.11809](https://arxiv.org/abs/2110.11809) (2021)
8. Cordeiro, F.R., Sachdeva, R., Belagiannis, V., Reid, I., Carneiro, G.: Longremix: robust learning with high confidence samples in a noisy label environment. Pattern Recogn. **133**, 109013 (2023)
9. Domingos, P.: A few useful things to know about machine learning. Commun. ACM **55**(10), 78–87 (2012)
10. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
11. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer. In: International Conference on Learning Representations (2016)
12. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer. In: International Conference on Learning Representations (2017)
13. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
15. Hendrycks, D., Mazeika, M., Wilson, D., Gimpel, K.: Using trusted data to train deep networks on labels corrupted by severe noise. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
16. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint [arXiv:1801.06146](https://arxiv.org/abs/1801.06146) (2018)
17. Iscen, A., Valmadre, J., Arnab, A., Schmid, C.: Learning with neighbor consistency for noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4672–4681 (2022)
18. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning, pp. 2304–2313. PMLR (2018)
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
21. 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)
22. 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)
23. Liu, Y., Cheng, H., Zhang, K.: Identifiability of label noise transition matrix. In: International Conference on Machine Learning, pp. 21475–21496. PMLR (2023)
24. 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)
25. Onal, K.D., et al.: Neural information retrieval: at the end of the early years. Inf. Retrieval J. **21**, 111–182 (2018)
26. Rong, Q., Yuan, L., Li, G., Li, J., Zhang, L., Ding, X.: A static bi-dimensional sample selection for federated learning with label noise. In: Wang, X., et al. (eds.)

- DASFAA 2023. LNCS, vol. 13943, pp. 735–744. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30637-2_49
- 27. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
 - 28. Smart, B., Carneiro, G.: Bootstrapping the relationship between images and their clean and noisy labels. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5344–5354 (2023)
 - 29. Sun, Z., Liu, H., Wang, Q., Zhou, T., Wu, Q., Tang, Z.: Co-LDL: a co-training-based label distribution learning method for tackling label noise. *IEEE Trans. Multimedia* **24**, 1093–1104 (2021)
 - 30. Tan, C., Xia, J., Wu, L., Li, S.Z.: Co-learning: learning from noisy labels with self-supervision. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 1405–1413 (2021)
 - 31. Wang, H., Xiao, R., Dong, Y., Feng, L., Zhao, J.: Promix: combating label noise via maximizing clean sample utility. *arXiv preprint arXiv:2207.10276* (2022)
 - 32. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 322–330 (2019)
 - 33. 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)
 - 34. Wei, H., et al.: Logit clipping for robust learning against label noise. *arXiv preprint arXiv:2212.04055* (2022)
 - 35. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with noisy labels revisited: a study using real-world human annotations. In: International Conference on Learning Representations (2022). <https://openreview.net/forum?id=TBWA6PLJZQm>
 - 36. Xia, X., et al.: Are anchor points really indispensable in label-noise learning? In: Advances in Neural Information Processing Systems, vol. 32 (2019)
 - 37. 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)
 - 38. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7017–7025 (2019)
 - 39. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning, pp. 7164–7173. PMLR (2019)
 - 40. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **64**(3), 107–115 (2021)
 - 41. Zhang, Y., Niu, G., Sugiyama, M.: Learning noise transition matrix from only noisy labels via total variation regularization. In: International Conference on Machine Learning, pp. 12501–12512. PMLR (2021)

Information Retrieval



A New Learning-to-Rank Framework for Keyphrase Extraction Using Multi-scale Ratings and Feature Fusion

Corina Florescu¹, Avijeet Shil²✉, and Wei Jin²✉

¹ Allstate - AI Center of Excellence, Dallas, TX, USA
CorinaFlorescu@my.unt.edu

² University of North Texas Discovery Park, 3940 N Elm Street, Denton, TX 76207, USA
AvijeetShil@my.unt.edu, Wei.Jin@unt.edu

Abstract. Previous work has mainly framed keyphrase extraction (KE) as a binary classification task where candidate phrases are predicted as either keyphrases or non-keyphrases. However, in reality, the boundary between them is somewhat hard to define according to a binary judgment, even for human annotators. Therefore, a finer measurement of appropriateness may be desired for this task, leading to our new idea of incorporating the degree to which a phrase represents the main topics of a document into the learning and ranking process. In this paper, we propose *ppKE*, a first supervised ranking model for keyphrase extraction that incorporates phrase importance information. A comprehensive feature study and evaluation are also conducted. Our model obtains remarkable improvements in performance over ranking models that do not take phrase relevance into account, as well as over strong previous approaches for this task.

Keywords: Keyphrase Extraction · Supervised Machine Learning · Feature Fusion · Learning-to-rank method

1 Introduction

Keyphrase extraction (KE) is the task of automatically extracting descriptive phrases or concepts that represent the main topics of a document. Keyphrase extraction is an important problem in Natural Language Processing (NLP), with applications ranging from search and retrieval to summarization, clustering, recommendation, online advertisement, and opinion mining [1–3]. KE has been intensively studied by employing various text corpora (e.g., research papers, news articles, tweets) and machine learning approaches, with a focus on the domain of academic papers and two lines of research: supervised and unsupervised [4].

Unsupervised keyphrase extraction is formulated as a ranking problem, where each candidate word of a target document receives a score based on various measures such as *tf-idf* [5], topic proportions, or PageRank [6,7]. Unsupervised

keyphrase extraction has been shown to work well in practice, with graph-based ranking methods being considered state-of-the-art. However, these methods compute a single importance score for each candidate word and are not flexible enough to incorporate multiple types of information.

Supervised keyphrase extraction is formulated as a classification problem where candidate phrases are classified as either positive (i.e., keyphrases) or negative (i.e., non-keyphrases). In contrast to unsupervised approaches, supervised models allow for integrating a myriad of information and techniques, and various feature sets and classification algorithms yield different extraction systems. However, formulating keyphrase extraction as a classification problem has several drawbacks. For example, a classifier identifies keyphrases independently of the other candidates in the document. Consequently, it cannot determine which candidates are more relevant to the topic of a given document. In contrast to other classification tasks where an instance intelligibly belongs to a particular class, the goal of keyphrase extraction is to identify the most representative phrases for a document rather than classifying phrases in an absolute sense.

To address this problem, Jiang et al. [8] proposed a ranking approach to keyphrase extraction, where the goal is to learn a model to rank two candidate keyphrases. Specifically, they employed a ranking model to sort keyphrases based on preference relations, where the order is induced by giving a binary judgment (e.g., “keyphrase” or “non-keyphrase”) for each phrase. We posit that not all keyphrases in a document have the same level of relevance to the topic of the document. Figure 1 shows an anecdotal example illustrating this behavior using a news article from one of our datasets. We can see that keyphrases such as “Ben Johnson”, “drug use”, and “Olympics” express the essence of the text and have a higher degree of importance compared with “100-meter dash” and “stanozolol”. More precisely, the article describes an incident of drug use involving Ben Johnson that caused him to leave the Olympics. However, it is less relevant what type of drug he used (e.g., stanozolol). This example shows that there are many boundary cases, and these instances are hard to classify even by humans. For example, Sterckx et al. [9] showed that the keyphrase agreement between different annotators is typically very small. We hypothesize that incorporating a phrase’s degree of relevance in the extraction process yields a more accurate list of keyphrases and has the potential to improve the keyphrase extraction task. To the best of our knowledge, this is the first approach to keyphrase extraction that incorporates a phrase preference score in the extraction process.

Our contributions are as follows:

- We propose a novel approach to keyphrase extraction using the preference attachment of a phrase. In particular, we effectively leverage the importance of keyphrases in the document to build a ranking model for extracting keyphrases.
- We experimentally validate our algorithm on several representative datasets and show statistically significant improvements over existing state-of-the-art models for keyphrase extraction.

Canadian **Ben Johnson** left the **Olympics** today “in a complete state of shock,” accused of cheating with drugs in the world’s fastest **100-meter dash** and stripped of his **gold medal**. The prize went to American **Carl Lewis**. Many athletes accepted the accusation that Johnson used a muscle-building but dangerous and illegal anabolic steroid called **stanozolol** as confirmation of what they said they know has been going on in track and field. Two tests of Johnson’s urine sample proved positive and his denials of **drug use** were rejected today. “This is a blow for the Olympic Games and the Olympic movement,” said International Olympic Committee President Juan Antonio Samaranch.

Fig. 1. An example of a document in our dataset and its human-annotated keyphrases

- Additionally, as part of our contributions, we annotated two benchmark datasets with phrase importance and we will make them available for further research in the keyphrase extraction community.

The rest of the paper is organized as follows. We summarize related work in the next section. We present our datasets in Sect. 3. A brief overview of learning to rank is provided in Sect. 4. Our proposed approach is described in Sect. 5. Section 6 presents our experiments and results. We conclude the paper in Sect. 7.

2 Related Work

Previous studies on keyphrase extraction have predominantly framed the task as a classification problem, employing supervised, unsupervised, and deep learning approaches.

In unsupervised approaches, measures such as *tf-idf* and topic proportions are used to score words, which are later aggregated to obtain scores for phrases [10, 11]. The ranking based on *tf-idf* has proved to work well in practice [5], despite its simplicity. Graph-based ranking methods and centrality measures are considered state-of-the-art. Since their introduction, many graph-based extensions have been proposed, which aim at modeling various types of information [12–15]. Several unsupervised approaches leverage word clustering techniques to group candidate words into topics, and then extract one representative keyphrase from each topic [16, 17]. Some approaches leverage sentence embedding on a single document and masking strategy to find the semantic similarity between documents and rank the candidate words [18, 19]. Recent state-of-the-art statistical approaches, exemplified by YAKE [20], incorporate a multifaceted analysis of keyword candidates. In addition to frequency-based features, these methods integrate factors such as casing, position, and contextual relevance to find the score for each candidate keyword.

Table 1. Summary of our datasets

Dataset	# Phrases	# Labeled 0	# Labeled 1	# Labeled 2	# Labeled 3
DUC	43,068	38,329	2,296	1,476	967
Inspec	9,346	6,338	1,448	897	663
DUC + Inspec	52,414	44,667	3,744	2,373	1,630

In supervised approaches, KE is commonly formulated as a binary classification problem where various feature sets and classification algorithms generally produce different models. For example, KEA [21], a representative supervised approach for KE, extracts two features for each candidate phrase, i.e., the *tf-idf* of a phrase and its distance from the beginning of the document, and uses them as input to Naïve Bayes. Hulth [22] used a combination of lexical and syntactic features such as the collection frequency, the relative position of the first occurrence, and the part-of-speech tag of a phrase in conjunction with a bagging technique. Medelyan [23] extended KEA to incorporate various information from Wikipedia. Structural features (e.g., the presence of a phrase in particular sections of a document) have been extensively used for extracting keyphrases from research papers [24, 25]. Koloski et al. [26] introduced a TF-IDF tagset matching method to discover and prioritize keywords in news articles across less-resourced languages such as Croatian, Latvian, Estonian, and well-resourced languages like Russian. Monali et al. [27] proposed the “Supervised Cumulative TextRank” where they introduced statistically supervised weight to incorporate the class label information of the terms.

In deep learning approaches, Meng et al. [28] proposed a generative model for predicting keyphrases for capturing the deep semantic meaning of the content. Zhang et al. [29] proposed a deep recurrent neural network model for predicting keyphrases from the social media platform Twitter.

In this paper, we are adopting the Learning-to-Rank technique (KE) to solve this from a ranking perspective, assigning different weights to phrases based on their importance in the text. Prior works on learning-to-work are conducted as both pointwise and pairwise ranking. A generic algorithm ranks the phrases based on their position and frequency with pointwise ranking [30]. Jiang et al. [8] used a set of traditional features in conjunction with the pairwise learning-to-rank technique to design a ranking approach to KE. Several works adopt a co-training framework in KE to solve model inconsistency and improve iteration time [31, 32].

In this paper, we propose a new approach following the learning-to-rank paradigm that attempts to incorporate a word or phrase’s degree of relevance to the topic of a document into the keyphrase identification task. A multi-scale rating scheme has been designed and used in the evaluation. To the best of our knowledge, this is the first approach to keyphrase extraction that leverages a phrase preference score in the extraction process.

In probability theory¹ and related fields⁰, a Markov process³, named after the Russian mathematician Andrey Markov¹, is a stochastic process² that satisfies the Markov property² [...]. A Markov chain³ is a type of Markov process³ that has either discrete state space³ or discrete index set³, [...]. Random walks¹ on integers⁰ and the gambler's ruin problem¹ are examples⁰ of Markov processes³[...]. Markov chains³ have many applications⁰ as statistical models¹ of real-world processes⁰, [...]. The algorithm known⁰ as PageRank¹, which was originally proposed for the internet search engine Google⁰, is based on a Markov process³[...]

Fig. 2. An example of an annotated document based on our scoring scheme

3 Data

Existing benchmark datasets for keyphrase extraction do not contain phrase relevance information. To train our model, we need gold-standard annotated phrases and their associated importance for the main topics of the target document. We built two datasets by annotating two existing collections (DUC and Inspec, described in Table 1) with phrase relevance information. Annotating documents with phrase relevance is a time-consuming process that is difficult to achieve for a collection like Inspec, which contains 2000 documents. Hence, we randomly selected 300 documents from these two collections and annotated them with phrase importance, as explained below. For each document, we extracted all phrases following the pattern (adjective)*(noun)+, and then assigned an integer score ranging from 0 to 3 to each phrase. This score indicates the level of relevance to the document's topics, with 0 indicating no relevance (i.e., a non-keyphrase) and 3 indicating a strong connection to the main topic. Below is a detailed explanation of the score scale:

- **0** (zero) or *not related* - the phrase does not represent the topics of that document;
- **1** or *weakly related* - the phrase is weakly related to the document;
- **2** or *related* - the phrase links to at least one of the topics of the document;
- **3** or *strong related* - the phrase represents the main topic of the document;

Figure 2 shows an example of an annotated document based on the scoring scheme presented above. The phrases in black represent candidate phrases in that document, while the grayed-out phrases are formed with words that do not pass the part-of-speech filters and consequently are removed from the set of candidates. The figure presents the assigned score for each candidate phrase as an exponent.

From this figure, we can see that our annotations provide more information on the redundant phrases. We label the “Markov chain” and “Markov process” with 3 to promote that the two phrases are equivalent. On the other hand, a phrase like “Andrey Markov” receives a score of 1 to distinguish that even if this phrase contains an important word (i.e., the central word in the constructed document graph-Markov), it does not represent the main topic of the text. Note that over-generation is a common type of error in the keyphrase extraction task where a system correctly predicts a candidate as a keyphrase because it contains

a word that appears frequently in the associated document, but at the same time erroneously outputs other candidates as keyphrases because they contain the same word.

We annotated the datasets with the help of two graduate students who were carefully instructed to assign the scores. We chose not to use crowd-sourcing services because the task requires attention and consideration, which hired annotators may not be willing to provide. Each document was annotated by both students, and the disagreements between them were solved through discussions instead of taking the mean of their assigned scores. The reason for this is that our target variables exist on an arbitrary scale where only the relative ordering between different values is significant. A summary of our datasets is presented in Table 1. The table shows, for each collection, the number of phrases considered for annotation and statistics on the scores they received. The last row in the table shows the statistics for the combined dataset, which will be used for some of the experiments presented in Sect. 6.

4 Learning to Rank

Different from classification and regression, the goal of learning to rank is to automatically construct a ranking model using training data, such that the model can rank instances according to their degree of preference, importance, or relevance. Learning to rank methods was initially applied in information retrieval, but recently it has shown good results in other areas such as machine translation, computational biology, and recommendation systems [33]. Learning to rank differs from regression and classification in the sense that it does not need to predict the absolute value or the absolute class of an instance. The relative ranking of items is all that is important. There are three common approaches to learning to rank: pointwise, pairwise, and listwise. The pointwise approach tries to directly apply existing learning models. That is, the ranking function learns to predict an absolute score for each instance in isolation, although this may not be necessary when the target is to produce a ranked list of the items. The pairwise approach no longer assumes absolute relevance, but it cares about the relative order between two documents. Specifically, the pairwise approach reduces the problem to a binary classification task, where the goal is to minimize the number of misclassified instance pairs. Listwise approaches aim to optimize the appropriate evaluation metric, directly averaging over all examples in the training data. We use the pairwise approach to turn our binary classification problem into a ranking problem. We chose pairwise over the pointwise approach because the former works on pairs of instances instead of isolated feature vectors. This is an important aspect of keyphrase extraction since keyphrases arise from a group of words whose degree of relevancy to the topic of the document is important rather than simply having the property of being keyphrases.

4.1 The Pairwise Approach

Given lists of items with some order between items in each list as a training corpus, the pairwise approach includes a preference constraint for all pairs of examples in the training corpus, for which the target value differs. Two examples are considered for a pairwise preference constraint only if they appear within the same list. Let us consider two lists of items $L_1 = \{i_{11}, i_{12}, i_{13}\}$ and $L_2 = \{i_{21}, i_{22}, i_{23}, i_{24}\}$ and their associated target values $T_1 = \{0, 0, 1\}$ and $T_2 = \{2, 3, 0, 0\}$. Then we can form the following pair constraints: $\{(i_{11}, i_{13}), (i_{12}, i_{13}), (i_{21}, i_{22}), (i_{21}, i_{23}), (i_{21}, i_{14}), (i_{22}, i_{23}), (i_{22}, i_{24})\}$. These preferences are then used as examples for a binary classifier that can tell which item is better in a given pair of items. Specifically, if we consider a linear ranking function, the ranking problem can be transformed into a two-class classification problem. Let us consider $X \subset \mathbb{R}^n$ the set of input feature vectors, $Y = \{y_1, y_2, \dots, y_m\}$ the target ranks and f a linear ranking function such as $f = w^T x$. Then

$$\begin{aligned} f(x_i) > f(x_j) &\iff w^T x_i > w^T x_j \\ &\iff w^T (x_i - x_j) > 0 \iff y_i > y_j \end{aligned}$$

where w is a weight vector, $x_i, x_j \in X$ and $y_i, y_j \in Y$. Then $x_i - x_j$ can be considered as a positive example and assign +1 if $x_i > x_j$ or as a negative example (-1) otherwise. Any binary classifier can be used to learn the weight vector w although SVM^{Rank} is a common choice. SVM^{Rank} and SVM have the same objective function, the difference lies in the constraints, which are constructed from pairs of instances.

4.2 RankNet

Several algorithms can be used to learn the ranking function with SVM^{Rank} [34] being the most popular. We chose to explore RankNet [35] for our task since it is usually implemented using a neural network which is more flexible and can approximate more complex bounded continuous functions. We implemented RankNet using a feed-forward neural network with two hidden layers of ten neurons each (more details on the neural network's architecture are provided in Sect. 6). RankNet maps an input feature vector $x \in R^n$ to a number $f(x)$. Specifically, each pair $(x_{i,d}, x_{j,d})$ with different labels is presented to the model which computes a feed-forward propagation for each training example and gives the scores $s_i = f(x_{i,d})$ and $s_j = f(x_{j,d})$. Let us assume that $x_{i,d}$ should be ranked higher than $x_{j,d}$. Then, the two outputs of the model are mapped to a learned probability that $x_{i,d}$ should be ranked higher than $x_{j,d}$ using a sigmoid function, as follows:

$$P_{ij} = \frac{e^{(s_i - s_j)}}{1 + e^{(s_i - s_j)}}$$

We then apply the cross-entropy cost function, which penalizes the deviation of the model's output probabilities from the desired probabilities

$$C = \bar{P}_{ij} \log(P_{ij}) - (1 - \bar{P}_{ij}) \log(1 - P_{ij})$$

where \bar{P}_{ij} is the known probability that the training example $x_{i,d}$ should be ranked higher than the training instance $x_{j,d}$.

To update the weights $w_k \in R$ (i.e. the model parameters) we use stochastic gradient descent:

$$w_k \leftarrow w_k - \eta \frac{\delta C}{\delta w_k} \leftarrow w_k - \eta \left(\frac{\delta C}{\delta s_i} \frac{\delta s_i}{\delta w_k} + \frac{\delta C}{\delta s_j} \frac{\delta s_j}{\delta w_k} \right)$$

where η is the learning rate. Thus, training RankNet is accomplished by a straightforward modification of back-propagation.

At the testing time, we use the trained ranking model to assign a score to each candidate phrase of a target document. Phrases are sorted by their scores and top k are output as the keyphrases of that document.

5 Proposed Model

Our phrase-preference keyphrase extraction (ppKE) model is a supervised ranking model, built on a combination of features described below. *ppKE* ranks candidate phrases using a slightly modified version of the RankNet algorithm and is implemented using a feed-forward neural network.

Table 2. An example of data transformation

(a)			(b)				
Data	Features	Relevance	Data	Features	Relevance		
x_{1,d_1}	0.343 0.237	0.018	$x_{1,d_1} - x_{2,d_1}$	-0.228 0.124	-0.855	2	(3-1)
x_{2,d_1}	0.571 0.113	0.873	$x_{1,d_1} - x_{3,d_1}$	-0.113 0.145	-0.016	1	(3-2)
x_{3,d_1}	0.456 0.092	0.034
x_{4,d_1}	0.620 0.022	0.378
x_{1,d_2}	0.456 0.092	0.034	$x_{4,d_1} - x_{2,d_1}$	0.049 -0.091	-0.495	1	(2-1)
x_{2,d_2}	0.620 0.022	0.378	$x_{1,d_2} - x_{2,d_2}$	0.049 -0.091	-0.495	-1	(1-2)

5.1 Features

We consider the following features in our model, which are described below.

Phrase's Tf-Idf. This feature represents the term frequency-inverse document frequency of a candidate phrase, computed for each target document. The “idf” component was estimated from Wikipedia which computes the inverse document frequency from the training set. We chose to use Wikipedia because we aim to

design a domain-independent model for keyphrase extraction. TF-IDF has been computed based on the following formula:

$$TF - IDF(word) = \#(word, doc) * \log\left(\frac{N}{1 + n_{word}}\right)$$

Phrase's First Position. This feature is computed as the position of the first occurrence of a phrase normalized by the number of tokens of the target document. This feature was first used by [36] and has been extensively explored since then using both supervised and unsupervised approaches [15, 22, 37].

Spread. Spread refers to the distance (computed in a number of tokens) between the first and last occurrences of a phrase in the target document. Similarly to the phrase's first position, the spread is normalized relative to the length of the document.

Phrase's Length. This feature captures the length of a phrase as the number of tokens. This feature has previously been used by [23] to capture the specificity of a phrase.

PageRank. The PageRank score obtained on a word graph built from adjacent words in a document has been used in the unsupervised line of research to rank phrases and select top k as keyphrases for the target document. However, the unsupervised approaches rely solely on this score to output keyphrases, we use the PageRank score of a phrase as a feature and aim at combining its strength with that of other features.

Word-Document Similarity. This feature reflects the similarity of a word to the target document computed in terms of word embeddings [38]. Distributed representations of words or “word embeddings” are models that represent the words in a low-dimensional space, each word being “embedded” as a d -dimensional vector of real numbers. These embeddings constitute an efficient representation of the text and have been shown to capture certain aspects of similarity between words without human intervention or language-dependent processing. We trained the Word2Vec model [38] on Wikipedia to obtain the embedding of words. Then, we computed the embedding of a document by taking the component-wise mean of the vectors of words constituting the document. However, not all words in a document have the same importance to the topic of that document and a simple weighted sum will just flatten the meaning of the document.

The weight most used in Information Retrieval (IR) is the TF-IDF of a word and it has been shown to work very well at capturing various aspects of information in a document. Hence, we compute the embedding of a document as a TF-IDF weighted component-wise mean of the vectors of words.

$$v_{doc} = \sum_{word \in doc} \frac{TF - IDF(word) * v_{word}}{|doc|}$$

where $TF-IDF(word)$ is the term-frequency-inverse document frequency of word $word$, v_{word} is the word embedding of $word$ and $|doc|$ is the number of words in the document doc .

The similarity between a word and the target document is computed as the cosine similarity between the two vectors as follows:

$$sim(word, doc) = \frac{v_{word} * v_{doc}}{\|v_{word}\| * \|v_{doc}\|}$$

This feature reflects that the more similar a word/phrase is to the “center of the document”, i.e., the document embedding the more likely it is to be a keyphrase.

Other Features. Our model is constructed based on six features outlined earlier. Additionally, we explored topic features, including word probability within document topics, total correlation to gauge token dependency within phrases, and node features derived from the word graph (e.g., node degree). These topics were inferred using Latent Dirichlet Allocation trained on Wikipedia data. However, these additional features did not significantly enhance our model’s performance.

5.2 Learning-to-Rank Keyphrases

We formulate our keyphrase extraction problem as follows. Let $D = \{d_1, d_2, \dots, d_p\}$ be a collection of documents where p is the total number of documents. We assume that each document $d \in D$ has a set of candidate keyphrases $C_d = \{c_{1,d}, c_{2,d}, \dots, c_{n,d}\}$, and each $c_{i,d} \in C_d$ is associated with a rank that reflects the degree to which $c_{i,d}$ is relevant to the topic of document d . Let $L = \{l_1, l_2, \dots, l_m\}$ be the set of all relevant degrees (levels) such that there is a total order among them. We denote with n the total number of candidate keyphrases for document d and with m the number of preference levels. We denote by $x_{i,d}$ the feature vector of candidate $c_{i,d}$, i.e., candidate i in document d , and by $l_{i,d}$ the target value of candidate $c_{i,d}$. During training, the target preference values are used to generate pairwise preference constraints. Two examples are considered for a pairwise preference constraint if (1) they appear within the same document; and (2) they are assigned different relevance ranks. Thus, the training data is formed of pairs of words and their associated label.

Table 2 shows a sample example of how we transform the previous training data to suit the setting of our new approach. As we can see in Table 2(b), we form the difference of all comparable elements in our data. Thus, the training data is formed of pairs $(x_{i,d} - x_{j,d}, l_{i,d} - l_{j,d})$, where $l_{i,d} = l_{i,d} - l_{j,d}$. The goal for the ranker is to minimize the number of inversions in ranking, i.e., cases where the pairs of results are in the wrong order relative to the ground truth.

Given a new document d , we ideally aim to return a permutation of its candidate phrases C_d that makes $c_i, i = 1, n$ have labels as ordered as possible, i.e., candidate phrases should be ranked based on their relevance to the topic of the document. Note that we can either consider $m = 2$ and the rank levels

as “keyphrase” and “non-keyphrase” or we can choose $m > 2$ and define some ordered categories that reflect a phrase relevant to the topic of the document, such as “strong related”, “related”, “weakly related” and “not related”.

The first case ($m = 2$) is trivial since we already have annotated datasets with gold standard keyphrases, and we can use these labels as ground truth [8]. For example, we can assign a score of 1 to all annotated keyphrases, while all the other phrases in the document receive a score of zero. However, to obtain a ranked list of keyphrases that better reflect their keyphraseness to a document, we need phrase relevance degree annotations. Existing benchmark datasets for keyphrase extraction do not contain such information. Hence, we plan to build a dataset of text documents such that each candidate phrase in a document is associated with a preference score, which indicates its informativeness on the topic of that document.

Several algorithms can be used to learn the ranking function, with SVM^{Rank} [34] being the most popular. We chose to explore RankNet [35] for our task since it is usually implemented using a neural network that is more flexible and can approximate more complex bounded continuous functions. We implemented RankNet using a feed-forward neural network with two hidden layers of ten neurons each (more details on the neural network’s architecture are provided in Sect. 6). At the testing time, we use the trained ranking model to assign a score to each candidate phrase of a target document and sort based on their scores, and the top k are output as the keyphrases of that document.

Table 3. The performance of *ppKE* in comparison with strong previous models

Approach	DUC			Inspec		
	P	R	F1	P	R	F1
Maui	0.261	0.336	0.290	0.205	0.250	0.211
RankingSVM	0.249	0.352	0.287	0.381	0.494	0.406
TopicRank	0.203	0.267	0.228	0.259	0.359	0.288
PositionRank	0.264	0.348	0.297	0.332	0.320	0.312
ppKE (binary)	0.299	0.412	0.341	0.369	0.478	0.395
ppKE (fine-grain)	0.311	0.429	0.355	0.392	0.511	0.420

6 Experiments and Results

6.1 Experimental Design

To evaluate our model, we randomly split each dataset into 75% training and 25% testing, ensuring instances from the same document remain together. Hyperparameters, including hidden layers, learning rate, regularization, and activation

function, were optimized using 10-fold cross-validation at the document level. Results are reported solely on the testing set.

We assess our model's performance using precision (P), recall (R), and F1-score (F1). Like our comparison models, we evaluate the top 10 predicted keyphrases generated by the model. Candidates are ranked based on either the confidence scores output by the classifier (for some previous approaches) or the scores of the ranking function.

Neural and **neuro-fuzzy integration** in a knowledge-based system for air quality prediction. We propose a unified approach for integrating implicit and explicit knowledge in neurosymbolic systems as a combination of neural and **neuro-fuzzy modules**. In the developed hybrid system, a training data set is used for building **neuro-fuzzy modules**, and represents implicit domain knowledge. The explicit domain knowledge on the other hand is represented by **fuzzy rules**, which are directly mapped into equivalent **neural structures**. This approach aims to improve the abilities of **modular neural structures**, which are based on incomplete learning data sets, since the knowledge acquired from human experts is taken into account for adapting the general neural architecture. Three methods to combine the explicit and implicit knowledge modules are proposed. The techniques used to extract **fuzzy rules** from neural implicit knowledge modules are described. These techniques improve the structure and the behavior of the entire system. The proposed methodology has been applied in the field of air quality prediction with very encouraging results. These experiments show that the method is worth further investigation.

Human-input keyphrases: *neuro-fuzzy integration, knowledge-based system, air quality prediction, neuro symbolic systems, hybrid system, training data set, fuzzy rules, incomplete learning, neural architecture, experiments*

Fig. 3. A document from the Inspec collection and the human-annotated keyphrases. Bold red phrases represent predicted keyphrases for the document (Color figure online)

6.2 Comparison with Baselines

We compare our model against supervised and unsupervised approaches for keyphrase extraction.

Supervised Approaches. Many supervised approaches for KE target specific corpora (to a large extent research papers) and rely on structural features that are not commonly available for all types of documents (e.g., citation contexts, sections). Since our data concerns both research papers and news articles, we compare SurfKE with two supervised models, Maui [23] and RankingSVM [8]. Maui includes features such as frequency and position of a phrase in a document, keyphraseness, the spread of a phrase in a document, and statistics gathered from Wikipedia such as the likelihood of a term is a link in Wikipedia or the number of pages that link to a Wikipedia page. Maui has been shown to perform well on various domains and documents. RankingSVM [8] is a learning-to-rank system for keyphrase extraction that leverages a set of standard features to train SVM^{Rank}.

Unlike our approach, they use only binary labels i.e., two phrases are considered for a pairwise constraint if one is annotated as a keyphrase and the other one as a non-keyphrase. Note that our method places constraints between two labeled keyphrases if one represents the topics of the document to a greater extent than the other one. These finer-grain constraints help the model better infer the most informative features for distinguishing keyphrases from non-keyphrases.

Unsupervised Approaches. Unsupervised approaches to KE have attracted significant attention since they typically do not require linguistic knowledge, nor domain-specific annotated corpora, which makes them easily transferable to other domains. Hence, we also compare *ppKE* with two unsupervised models, TopicRank [17] and PositionRank [15]. TopicRank groups candidate phrases into topics and uses them as vertices in a complete graph. The importance of each topic is computed using graph-based ranking functions, and keyphrases are selected from the highest-ranked topics. PositionRank is a graph-based model, that exploits the positions of words in the target document to compute a weight for each word. This weight is incorporated into a biased PageRank algorithm to score words that are later used to rank candidate phrases.

Table 3 shows the comparison of *ppKE* with the previous work described above. As we can see in the table, *ppKE* substantially outperforms both supervised and unsupervised systems on the two datasets. For instance, on DUC, *ppKE* gets relative improvements of 23.69%, 20.41%, and 19.52% over RankingSVM, Maui, and PositionRank, respectively, in terms of F1-score.

With a paired t-test, we found that the improvements in precision, recall, F1-score achieved by *ppKE* are all statistically significant with $p \leq 0.05$. The results presented in this paper are obtained using the authors' publicly available implementations for Maui and PositionRank and the implementation from the `pke` package [39] for TopicRank. Our code and data will be made freely available.

6.3 Cross-Domain Performance

Annotating documents with keyphrases is time-consuming, especially when assigning fine-grained scores to each phrase. In this experiment, we investigate whether a model trained on existing labeled data can effectively extract keyphrases from other document collections. To test this, we combined two annotated datasets (DUC and Inspec) and trained our model on the merged collection. Subsequently, we evaluated the model on a new dataset of research papers called NUS [24]. NUS comprises 211 research papers, with author-input keyphrases serving as the gold standard for evaluation, with an average of four keyphrases assigned to each paper.

Results on NUS. The results of this experiment, presented in Table 4, highlight the performance of our model compared to baseline methods. Notably, our model surpasses all baselines when evaluated on the NUS dataset. For example, *ppKE* achieves an F1-score of 0.172, outperforming RankingSVM and TopicRank with scores of 0.158 and 0.152, respectively.

Table 4. The performance of *ppKE* in a cross-domain setting

Approach	Precision	Recall	F1-score
Maui	0.090	0.388	0.142
RankingSVM	0.101	0.420	0.158
TopicRank	0.104	0.297	0.152
PositionRank	0.094	0.234	0.131
ppKE	0.110	0.451	0.172

It’s important to note that all models exhibit lower performance on the NUS dataset compared to DUC and Inspec. This could be attributed to the difference in keyphrase assignment processes. DUC and Inspec datasets have more keyphrases per document (around 10) and were annotated by readers instructed to select phrases present in the document. In contrast, keyphrases in the NUS dataset are provided by the authors, typically consisting of around three keyphrases per document, sometimes with equivalent phrases like “topic modeling” instead of “Latent Dirichlet Allocation.”

6.4 Anecdotal Evidence

We show anecdotal evidence using a document from the Inspec collection. Figure 3 shows the abstract of this paper together with the author-input keyphrases. We marked in bold dark red the candidate phrases that are predicted as keyphrases by our proposed model (*ppKE*). We can see from this example that the keyphrases outputted by our model are close to the human annotated for a given document. However, we can notice that our model tends to be redundant and finds it difficult to distinguish between phrases such as “modular neural structures” and “neural structures”.

7 Conclusion

This paper presents a novel ranking model for keyphrase extraction, integrating a fine-grained scale to evaluate a phrase’s relevance to the topics of the target document. Our approach involves assigning a score to each candidate phrase based on its significance to the document’s content. These scores are utilized within a pairwise learning-to-rank framework to train a ranking model capable of identifying the top k keyphrases for any given document.

Our experiments show that:

1. Our supervised model, *ppKE*, obtains remarkable improvements in performance over four previous proposed models for KE.
2. The performance of *ppKE* is consistent across domains, which makes it easily transferable to other domains.

In the future, we plan to experiment and evaluate the listwise approach, an alternative approach to learning to rank, for this task.

References

1. Xiong, A., Liu, D., Tian, H., Liu, Z., Peng, Yu., Kadoch, M.: News keyword extraction algorithm based on semantic clustering and word graph model. *Tsinghua Sci. Technol.* **26**(6), 886–893 (2021)
2. Yang, Y., Li, H.: Keyword decisions in sponsored search advertising: a literature review and research agenda. *Inf. Process. Manag.* **60**(1), 103142 (2023)
3. Hernández-Castañeda, Á., García-Hernández, R.A., Ledeneva, Y., Millán-Hernández, C.E.: Extractive automatic text summarization based on lexical-semantic keywords. *IEEE Access* **8**, 49896–49907 (2020)
4. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1262–1273 (2014)
5. El-Beltagy, S.R., Rafea, A.: KP-miner: participation in semeval-2. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 190–193. Association for Computational Linguistics (2010)
6. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411 (2004)
7. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic keyphrase extraction via topic decomposition. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 366–376 (2010)
8. Jiang, X., Hu, Y., Li, H.: A ranking approach to keyphrase extraction. In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 756–757. ACM (2009)
9. Sterckx, L., Demeester, T., Deleu, J., Develder, C.: When topic models disagree: keyphrase extraction with multiple topic models. In: *Proceedings of the 24th International Conference on World Wide Web Companion*, pp. 123–124. International World Wide Web Conferences Steering Committee (2015)
10. Merrouni, Z.A., Frikh, B., Ouhbi, B.: Automatic keyphrase extraction: a survey and trends. *J. Intell. Inf. Syst.* **54**, 391–424 (2020)
11. Zhang, C., Zhao, L., Zhao, M., Zhang, Y.: Enhancing keyphrase extraction from academic articles with their reference information. *Scientometrics* **127**(2), 703–731 (2022)
12. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: *Proceedings of the 28th American Association for Artificial Intelligence*, pp. 1629–1635 (2014)
13. Wang, R., Liu, W., McDonald, C.: Corpus-independent generic keyphrase extraction using word embedding vectors. In: *Software Engineering Research Conference*, p. 39 (2014)
14. Martinez-Romo, J., Araujo, L., Duque Fernandez, A.: Semgraph: extracting keyphrases following a novel semantic graph-based approach. *J. Assoc. Inf. Sci. Technol.* **67**(1), 71–82 (2016)
15. Florescu, C., Caragea, C.: Positionrank: an unsupervised approach to keyphrase extraction from scholarly documents. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1105–1115 (2017)
16. Gagliardi, I., Artese, M.T.: Semantic unsupervised automatic keyphrases extraction by integrating word embedding with clustering methods. *Multimodal Technol. Interact.* **4**(2), 30 (2020)

17. Bougouin, A., Boudin, F., Daille, B.: Topicrank: graph-based topic ranking for keyphrase extraction. In: International Joint Conference on Natural Language Processing (IJCNLP), pp. 543–551 (2013)
18. Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., Jaggi, M.: Simple unsupervised keyphrase extraction using sentence embeddings. arXiv preprint [arXiv:1801.04470](https://arxiv.org/abs/1801.04470) (2018)
19. Zhang, L., et al.: Mderank: a masked document embedding rank approach for unsupervised keyphrase extraction. arXiv preprint [arXiv:2110.06651](https://arxiv.org/abs/2110.06651) (2021)
20. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: YAKE! keyword extraction from single documents using multiple local features. Inf. Sci. **509**, 257–289 (2020)
21. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, pp. 668–673 (1999)
22. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 216–223 (2003)
23. Medelyan, O., Frank, E., Witten, I.H.: Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 1318–1327. ACL (2009)
24. Nguyen, T.D., Luong, M.-T.: Wingnus: keyphrase extraction utilizing document logical structure. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 166–169. Association for Computational Linguistics (2010)
25. Lopez, P., Romary, L.: HUMB: automatic key term extraction from scientific articles in grobid. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 248–251. Association for Computational Linguistics (2010)
26. Koloski, B., Pollak, S., Škrlj, B., Martinc, M.: Extending neural keyword extraction with TF-IDF tagset matching. arXiv preprint [arXiv:2102.00472](https://arxiv.org/abs/2102.00472) (2021)
27. Bordoloi, M., Chatterjee, P.C., Biswas, S.K., Purkayastha, B.: Keyword extraction using supervised cumulative textrank. Multimedia Tools Appl. **79**(41–42), 31467–31496 (2020)
28. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. arXiv preprint [arXiv:1704.06879](https://arxiv.org/abs/1704.06879) (2017)
29. Zhang, Q., Wang, Y., Gong, Y., Huang, X.-J.: Keyphrase extraction using deep recurrent neural networks on twitter. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 836–845 (2016)
30. Turney, P.D.: Learning to extract keyphrases from text. arXiv preprint [arxiv:cs/0212013](https://arxiv.org/abs/cs/0212013) (2002)
31. Zhao, C., Yan, J., Liu, N.: Improve web search ranking by co-ranking SVM. In: 2008 Fourth International Conference on Natural Computation, vol. 2, pp. 81–85. IEEE (2008)
32. Wang, C., Li, S.: Corankbayes: Bayesian learning to rank under the co-training framework and its application in keyphrase extraction. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2241–2244 (2011)
33. Zehlike, M., Yang, K., Stoyanovich, J.: Fairness in ranking, part II: learning-to-rank and recommender systems. ACM Comput. Surv. **55**(6), 1–41 (2022)
34. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)

35. Burges, C., et al.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. ACM (2005)
36. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: IJCAI **1999**, 668–673 (1999)
37. Caragea, C., Bulgarov, F., Godea, A., Gollapalli, S.D.: Citation-enhanced keyphrase extraction from research papers: a supervised approach. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1435–1446 (2014)
38. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
39. Boudin, F.: PKE: an open source python-based keyphrase extraction toolkit. In: COLING 2016, pp. 69–73 (2016)



MIIGraph: Multi-granularity Information Integration Graph for Document-Level Event Extraction

Lin Mu, Yide Cheng, Xiaoyu Wang, Yang Li, and Yiwen Zhang^(✉)

Anhui University, Hefei, China

{mulin, Zhangyiwen}@ahu.edu.cn,

{yide_cheng, e23301216, g12114008}@stu.ahu.edu.cn

Abstract. Document-level Event Extraction (DEE) involves extracting event-related structural information, such as event types and event arguments, from a document containing multiple sentences. This task presents challenges, including argument scattering, multiple events, and role overlap, compared to Sentence-level Event Extraction (SEE). Existing works construct heterogeneous graphs for DEE to capture the interactions between entities and sentences. However, they neglect the importance of the global theme information and the interaction information between entities, sentences, and global theme information. To address this gap, we propose the **Multi-granularity Information Integration Graph (MIIGraph)** framework for DEE. This model aims to capture the interaction of multi-granularity information such as entities, sentences, and global theme of a document for DEE. Specifically, we first obtain the global theme representation of the document through contrastive learning. Then, we construct a heterogeneous graph to capture the complex interactions between entities, sentences, and global theme. Finally, we conducted extensive experiments to evaluate MIIGraph on two widely used DEE benchmarks. The results show that MIIGraph significantly improves the performance of DEE compared to existing methods.

Keywords: Event Extraction · Multi-Granularity · Heterogeneous Graph · Contrastive Learning

1 Introduction

Event extraction (EE) [11] is a significant subtask in the field of information extraction (IE). The primary objective of EE is to identify and extract valuable information related to events from unstructured text. For example, as shown in Fig. 1, EE aims to extract the event type *EquityUnderweight* and event arguments *Mingting Wu*(role = *EquityHolder*), *7.2 million*(role = *Traded Shares*) from a text. Event extraction plays an indispensable role in facilitating a myriad of downstream tasks, such as knowledge graph construction [4], question answering systems [1], and recommendation systems [23].

Most of the previous methods [2, 16, 17] predominantly concentrated on event extraction from unstructured data at the sentence level, commonly referred to as sentence-level event extraction (SEE). However, real-world events are often described in multiple sentences, i.e., in documents [12]. The intricate relationships among entities and sentences in documents present multiple challenges for this task. Specifically, for document-level event extraction (DEE), there are the following challenges:

- **Argument Scattering:** The event arguments about an event are dispersed across multiple sentences. For instance, in Fig. 1, the event arguments for the *EquityUnderweight* event type are scattered throughout multiple sentences in the document.
- **Multiple Events:** A document may encompass multiple events. For example, in Fig. 1, this document contain two event types *EquityUnderweight* and *EquityOverweight* event type.
- **Role Overlap:** An event argument may correspond to multiple argument roles. For instance, in Fig. 1, *Nov 6, 2014* serves as both the *StartDate* and *EndDate* for the *EquityUnderweight* event type.

Document: [1] On *Nov 6, 2014*, the company received a letterof share reduction from *Mingting Wu*, the shareholder of the company. [2] *Mingting Wu* decreased his holding of *7.2 million* shares of the company on the Shenzhen Stock Exchange on *Nov 6, 2014*. [3] The *7.2 million* shares of the company *Mingting Wu* reduced this time were transferred to *Xiaoting Wu*. [4] *Xiaoting Wu* is the daughter of *Mingting Wu*, and they were identified as personsacting in concert according to relevant regulations.

DEE

EventType	EquityHolder	TradedShares	StartDate	EndDate	LaterHoldingShares	AveragePrice
EquityUnderweight	Mingting Wu	7.2 million	Nov 6, 2014	Nov 6, 2014	NULL	NULL
EquityOverweight	Xiaoting Wu	7.2 million	Nov 6, 2014	Nov 6, 2014	7.2 million	NULL

Fig. 1. An example of document-level event extraction using a document from the ChFinAnn dataset is illustrated. The entity mentioned within the document has been highlighted. The final results of the document-level event extraction are presented in the table at the bottom of the figure.

Previous studies [19, 25] have utilized heterogeneous graphs to capture the complex interactions between entities and sentences in documents, which can effectively alleviate the aforementioned challenges. However, they overlooked the importance of global theme information (a broad directional topic such as business, sports, technology, etc.) in DEE. For instance, the entity “*bank*” refers to a financial institution within a financial context, but it can also denote a riverbank

in a geographical setting. Extracting global theme information from the document and constructing the interaction between the global topic, entities, and sentences can better capture the semantic information of the document and improve the performance of DEE. To obtain global theme information, it is important to acquire the global semantic representation of the document. Through various text perturbations, contrastive learning can effectively grasp the core semantic information of text [6]. This is crucial for capturing the global theme information of the document. Therefore, in this paper, we use contrastive learning methods to obtain global theme information from documents and utilize this information to facilitate DEE.

In this paper, we propose a **Multi-granularity Information Integration Graph (MIIGraph)** framework for document-level event extraction. Specifically, first, we obtain the global theme representation of the document through contrastive learning. Second, we construct a heterogeneous graph to capture the complex interaction between entities, sentences, and global theme in documents. Finally, the heterogeneous graph is trained by a heterogeneous graph neural network, and event detection and event argument extraction are performed.

Our contributions include the following:

- We propose a framework called Multi-granularity Information Integration Graph (MIIGraph) for DEE. MIIGraph effectively alleviates several challenges in document-level event extraction and improves performance by integrating multi-granularity information such as entities, sentences, and global theme.
- To enhance global theme information extraction, we utilize the contrastive learning technique for DEE. This method enables acquiring high-quality global theme information from documents, consequently enhancing DEE performance.
- We conduct extensive experiments, and the results demonstrate that MII-Graph outperforms the baselines and achieves state-of-the-art performance. Specifically, we achieved absolute F1-Score improvements of 0.4% on ChFi-nAnn and 1.4% on DuEE, respectively.

2 Preliminaries

Firstly, we clarify several key concepts in event extraction tasks.

- **entity**: a real-world object, such as a person, organization, location, etc.
- **entity mention**: a text in the document that refers to the entity object.
- **argument role**: an attribute corresponding to a pre-defined field in an event.
- **event argument**: an entity playing a specific event role.
- **event record**: a record expressing an event itself, including a series of event arguments.

In a DEE task, a context may contain multiple event records, and an event record may not contain all the corresponding event arguments.

Following Doc2EDAG [24], DEE primarily consists of several tasks, including named entity recognition (NER), event detection (ED), and event argument extraction (EAE).

- **NER**: involves identifying the span of entities (e.g., “*Mingting Wu*”) within the document and assigning types to these entities (e.g., “*EquityHolder*”).
- **ED**: aims to identify specific types of events from documents.
- **EAE**: aims to identify various arguments (participants, roles, attributes, etc.) of events within the document and associate them with specific events.

3 Method

As shown in Fig. 2, MIIGraph can be divided into three components: 1) Extract the global theme information. 2) We then construct a heterogeneous graph to model the interaction between entity mentions, sentences, and global theme. 3) Finally, we do the final event detection and event argument extraction using the updated heterogeneous graph node properties.

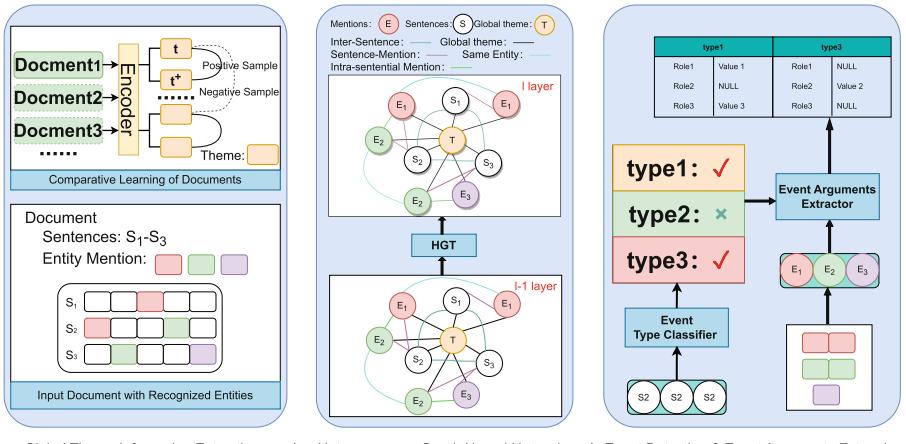


Fig. 2. Overview of MIIGraph. We first perform contrastive learning to obtain the global theme information of the document. Then, we model the document as a heterogeneous graph and use Heterogeneous Graph Transformer (HGTT) [7] to integrate multigranularity interaction information. Finally, we perform event detection and event argument extraction, yielding the final results.

3.1 Global Theme Information Extraction

We denote one document $d = \{s_1, \dots, s_i, \dots, s_N\}$ as a sequence of sentences, where s_i is i -th sentence in the d . Sentence $s = \{x_1, \dots, x_i, \dots, x_{|s|}\}$ as a

sequence of word, where x_i represents i -th word in the s , and $|s|$ denotes the length of the s , i.e., the number of words in the sentence s .

First, we employ contrastive learning to obtain high-quality global theme information from documents. Specifically, for each document d_i , we define a corresponding $d_i^+ = d_i$. Subsequently, d_i and d_i^+ are fed into two Transformers [18], which differ only in their dropout configurations, resulting in distinct document embedding t_i and t_i^+ . Other documents in the batch, processed through the Transformer to obtain embeddings denoted as t_j and t_j^+ , are considered negative samples concerning t_i , while t_i^+ serves as the positive sample. For training, we minimize the following loss function:

$$\mathcal{L}_{cont} = -\log \frac{\exp(sim(t_i, t_i^+)/\tau)}{\sum_{j=1}^{|t|} \exp(sim(t_i, t_j)/\tau)}, \quad (1)$$

where $sim(t_i, t_i^+)$ denotes the cosine similarity between the two samples t_i and t_i^+ , and τ is the temperature hyperparameter. This hyperparameter τ controls the smoothness of the distribution, and $|t|$ represents the number of documents in the batch.

Next, we move inside the document d_i and further refine the expression of the document. We apply max pooling to t_i to obtain the document's global theme information embedding $T = MaxPooling(t_i)$. Furthermore, we use the Transformer to encode each sentence s in the document to get a sequence of embeddings:

$$\{h_1, \dots, h_{|s|}\} = Transformer(\{x_1, \dots, x_{|s|}\}). \quad (2)$$

The word representation of x_j is a sum of the corresponding token and position embeddings.

For named entity recognition, we take a BIO (Begin, Inside, Other) schema for the sequence labeling task, which utilizes the Conditional Random Field (CRF) layer to recognize entities. The loss function is as follows:

$$\mathcal{L}_{ner} = -\sum_{s \in D} logP(y_s | s) \quad (3)$$

where y_s is the golden label sequence of s .

In addition, for each entity mention, we apply max pooling to all its tokens to obtain the entity mention embedding $\tilde{\mathbf{m}} = MaxPooling(x_i, \dots, x_j)$. For the sentence embedding, we also first apply max pooling to all its tokens and then concatenate it with the global theme information embedding T : $ss = concat((MaxPooling(x_1, \dots, x_{|s|}), T))$.

3.2 Heterogeneous Graph Neural Network

Event arguments in documents may be scattered across multiple sentences, which implies that the entity mentions corresponding to event arguments are distributed across different sentences. Therefore, cross-sentence modeling of these

entity mentions is the foundation for DEE. Furthermore, global theme information can provide guidance for DEE. Consequently, we construct a heterogeneous graph HG that incorporates global theme information nodes, entity mention nodes, and sentence nodes from the document d .

We introduce the following five types of edges in the heterogeneous graph to capture the interaction between entity mentions, sentences, and global theme information.

- **Global Theme Edge:** For global theme nodes, we leverage theme edges to connect them with all other nodes, thus facilitating a more comprehensive approach to our DEE process.
- **Inter-Sentence Edge:** Interconnecting all sentence nodes enables us to capture the global theme information of the document more effectively.
- **Sentence-Mention Edge:** We establish connections between all sentence nodes and nodes representing entity mentions within sentences, aiming to enhance the acquisition of sentence-level document context information.
- **Intra-sentential Mention Edge:** We establish connections among all entity mentions within the same sentence. Given that entities occurring within the same sentence are more likely to serve as arguments of the same event, we explicitly reinforce such relationships through this edge.
- **Same Entity Edge:** For distinct mentions of the same entity, it is imperative to establish connections among them to enhance the representation of the entity. Furthermore, sentences featuring identical entities are more likely to correspond to the same event [8], thus facilitating a more robust reinforcement of their relationship.

In the past, the majority of methodologies necessitated the design of metapaths for heterogeneous graphs, or they simply assumed either the sharing of identical features and representation spaces among different types of nodes/edges, or the maintenance of distinct non-shared weights solely for node types or edge types. This rendered them insufficiently equipped to fully capture the characteristics of heterogeneous graphs. Therefore, upon completing the heterogeneous graph construction, we employ the Heterogeneous Graph Transformer (HGT) [7] to model the global interactions within the heterogeneous graph. Given a target node t , evaluating the significance of all its neighbors (source nodes) $s \in N(t)$ involves computing the h-head attention for each edge $e = (s, t)$, expressed as:

$$\begin{aligned}
 \text{Attention}_{HGT}(s, e, t) &= \underset{s \in N(t)}{\text{Softmax}} \left(\left\| \underset{i \in [1, h]}{\text{ATT-head}^i(s, e, t)} \right\| \right) \\
 \text{ATT-head}^i(s, e, t) &= \left(K^i(s) W_{\phi(e)}^{ATT} Q^i(t)^T \right) \cdot \frac{\mu_{\langle \tau(s), \phi(e), \tau(t) \rangle}}{\sqrt{d}} \\
 K^i(s) &= \text{K-Linear}_{\tau(s)}^i \left(H^{(l-1)}[s] \right) \\
 Q^i(t) &= \text{Q-Linear}_{\tau(t)}^i \left(H^{(l-1)}[t] \right)
 \end{aligned} \tag{4}$$

where $\mathbf{W}_{\phi(e)}^{\text{ATT}} \in (\mathbb{R})^{\frac{d}{h} \times \frac{d}{h}}$, denotes the parameter matrix trained for each type of edge $\phi(e)$.

Subsequently, we propagate information from the source nodes to the target nodes. The specific process is outlined as follows:

$$\begin{aligned} \text{Message}_{HGT}(s, e, t) &= \bigcup_{i \in [1, h]} \text{MSG-head}^i(s, e, t) \\ \text{MSG-head}^i(s, e, t) &= \text{M-Linear}_{\tau(s)}^i \left(H^{(l-1)}[s] \right) W_{\phi(e)}^{MSG} \end{aligned} \quad (5)$$

where $\mathbf{W}_{\phi(e)}^{\text{MSG}} \in (\mathbb{R})^{\frac{d}{h} \times \frac{d}{h}}$, denotes the parameter matrix during the message passing phase for each type of edge $\phi(e)$.

We further utilize attention as weights to compute the average of the respective information from source nodes, yielding the updated embedding for the target node t :

$$\tilde{H}^{(l)}[t] = \bigoplus_{s \in N(t)} (\text{Attention}_{HGT}(s, e, t) \cdot \text{Message}_{HGT}(s, e, t)). \quad (6)$$

Finally, utilizing the target node's class $\tau(t)$ as an index, we map the embedding of the target node t back to the distribution corresponding to its respective class. Specifically, we apply a linear mapping $\text{A-Linear}_{\tau(t)}$ to the updated embedding $\tilde{H}^{(1)}[t]$, followed by the residual connection with the original embedding of node t from the previous layer:

$$H^{(l)}[t] = \text{A-Linear}_{\tau(t)} \left(\sigma \left(\tilde{H}^{(l)}[t] \right) \right) + H^{(l-1)}[t]. \quad (7)$$

Thus, we obtain the output $H^{(l)}[t]$ of the target node t in the l -th layer of HGT.

Finally, we obtain the global theme information embedding $TT \in \mathbb{R}^T$, the sentence matrix $SS = [H_1^T, \dots, H_{|s|}^T] \in \mathbb{R}^{d \times |s|}$, and the entity matrix $EE \in \mathbb{R}^{d \times |E|}$. Wherein the embedding of each entity is obtained by average-pooling the embeddings of all its entity mentions: $EE_i = \text{Mean}(\{H_j\}_{j \in \text{Mention}(i)})$

3.3 Event Detection and Event Arguments Extraction

Event Detection. Considering that a document may contain multiple different temporal types, we employ a binary classifier for each event type to predict whether the corresponding event is identified, instead of using Softmax regression. This is done to ensure the recognition of multiple events in the document as comprehensively as possible. Specifically, we utilize the sentence feature matrix SS for event detection tasks:

$$\begin{aligned} A &= \text{MultiHead}(Q, SS, SS) \in \mathbb{R}^{d_m \times |T|} \\ R &= \text{Sigmoid}(A^T W_t) \in \mathbb{R}^{|T|} \end{aligned} \quad (8)$$

where $Q \in \mathbb{R}^{d_m \times |T|}$ and $W_t \in \mathbb{R}^{d_m}$ are trainable parameters, and $|T|$ denotes the number of possible event types. MultiHead(\cdot) refers to the standard multi-head attention mechanism with Query/Key/Value. Therefore, we derive the event types detection loss with golden label $\hat{R} \in \mathbb{R}^{|T|}$:

$$\mathcal{L}_{ED} = - \sum_{t=1}^{|T|} \left(\mathbb{I}(\hat{R}_t = 1) \log P(R_t|D) + \mathbb{I}(\hat{R}_t = 0) \log(1 - P(R_t|D)) \right) \quad (9)$$

Event Arguments Extraction. We have now identified all potential events and entities contained within the document. The subsequent step involves integrating these elements to derive the final outcomes. In the task of role classification, an entity may assume multiple roles within a single record; however, each role within a table can only be occupied by one entity. Consequently, we adhere to the methodology proposed by GIT [19] for the extraction of event arguments.

For each record path, denoted as the i -th path, which comprises a sequence of entities, the Tracker transforms the associated sequence of entity representations $U_i = [E_{i1}, E_{i2}, \dots]$ into an embedding G_i utilizing an LSTM (capturing the last hidden state) and incorporates the event type embedding. This condensed record information is then stored in a global memory G , accessible across different event types. During the extraction process, for a record path $U_i \in \mathbb{R}^{d_m \times (J-1)}$ containing the first $J-1$ argument roles, the J -th role is predicted by augmenting the entity representations with role-specific information, $\tilde{EE} = EE + \text{Role}_J$, where Role_J represents the embedding for the J -th role. We subsequently concatenate \tilde{EE} , sentence features S , the current entity path U_i , and the global memory G . A transformer is then applied to generate a new entity feature matrix $\tilde{EE} \in \mathbb{R}^{d_m \times |E|}$, which integrates global role-specific information for all entity candidates:

$$[\tilde{EE}, \tilde{SS}, \tilde{U}_i, \tilde{G}] = \text{Transformer}([\tilde{EE}; \tilde{SS}; \tilde{U}_i; \tilde{G}])$$

Path expansion is approached as a multi-label classification challenge, where a binary classifier is applied to \tilde{EE}_i to determine whether the i -th entity correctly represents the next argument role for the current record, leading to an expansion of the path.

During training, the loss function is defined as:

$$\mathcal{L}_{EAE} = - \sum_{n \in N_D} \sum_{t=1}^{|E|} \log P(y_t^n | n)$$

Here, N_D signifies the set of nodes in the event records tree, and y_t^n represents the correct label. If the t -th entity is appropriate for the next argument in node n , then $y_t^n = 1$; otherwise, $y_t^n = 0$.

3.4 Optimization

We employ a joint training strategy [3] to integrate the loss functions of several distinct subtasks, formulated as:

$$\mathcal{L} = k_1 \mathcal{L}_{cont} + k_2 \mathcal{L}_{ner} + k_3 \mathcal{L}_{ED} + k_4 \mathcal{L}_{EAE} \quad (10)$$

where k_i represent hyperparameters aimed at balancing the weights of various components.

4 Experiment

4.1 Setup

DataSets. In the realm of DEE, two pivotal datasets, namely **ChFinAnn** [24] and **DuEE** [13], are considered essential resources driving research in this field.

- **ChFinAnn** is the largest DEE dataset constructed through distantly supervised alignment. It includes 32,000 financial announcements and 48,000 event records. The dataset is quite complex, with 29% of documents containing multiple records, and an impressive 98% of records having arguments distributed across multiple sentences. On average, each document in this dataset has 20 sentences, with the longest document containing 6,200 Chinese characters. The size and intricacy of ChFinAnn make it a cornerstone in the landscape of DEE datasets.
- **DuEE** is a comprehensive dataset that has been specifically designed for DEE. It has 19,640 events that are categorized into 65 distinct event types, and 41,520 event arguments that are mapped to 121 argument roles. The dataset is the most extensive Chinese EE dataset to date. The human annotation process was meticulous and involved crowdsourced reviews, ensuring that the annotation accuracy was over 95. The dataset’s schema comprises trending topics from Baidu Search, and the data is sourced from Baijiahao news articles, thus grounding the tasks in real-world scenarios. This makes DuEE an essential asset for researchers who are engaged in DEE pursuits, given its richness and diversity.

Experiment Settings. In our implementation, we utilize an 8-layer Transformer in the encoder module and a 4-layer Transformer in the decoder module. The contrastive learning segment employs the representation of the [CLS] token from the top MLP layer as the sentence representation. The dimensions of the hidden layers and feedforward layers remain consistent with prior work, namely 768 and 1,024, respectively. Additionally, we incorporate a 3-layer HGT with a dropout rate set to 0.2 and a batch size of 64. Training is performed using the Adam optimizer [9] with a learning rate of $3e^{-5}$ for 100 epochs. For the loss function, we set $k_1 = 0.5$, $k_2 = 0.05$, $k_3 = k_4 = 1$.

Baseline

- **DCFEE** [20]: The DCFEE utilized an argument-completion strategy to create a detailed event record for the whole document. This was achieved by leveraging arguments derived from the results of sentence-level event extraction. Two variants were introduced to address the challenge of multi-event extraction, namely DCFEE-O, and DCFEE-M. DCFEE-O was designed to generate a singular event record, while DCFEE-M focused on producing multiple potential argument combinations from a key-event sentence.
- **Doc2EDAG** [24]: Generated an entity-based directed acyclic graph to extract multiple events from documents. Additionally, the GreedyDec algorithm greedily populates one event table entry using recognized entity roles, sharing an identical structure with Doc2EDAG.
- **GIT** [19]: A new model called GIT was developed, which used a heterogeneous graph interaction network as an encoder and maintains a global tracker during the decoding process. GIT was an improvement over the Doc2EDAG model, incorporating a graph neural network to enhance entity encoding and introduce additional features during directed acyclic graph (DAG) generation.
- **PTPCG** [25]: A new approach was utilized to combine event arguments and a non-autoregressive decoding algorithm was applied using pruned complete graphs.
- **ReDEE** [14]: ReDEE was the first to model entity relationship information and guides DEE tasks through a novel relation-enhanced attention transformer.
- **ChatGPT**: A direct input of textual data and task parameters into ChatGPT has yielded results in event extraction [5].

4.2 Main Results

In this study, we introduce a multi-granularity information integration graph (MIIGraph) framework, which uses contrastive learning to model global theme information of documents and then integrates multi-granularity information by constructing a heterogeneous graph. To evaluate the effectiveness of MIIGraph, we conducted comprehensive experiments on two benchmark datasets, ChFinAnn and DuEE. The performance comparison with existing methods is summarized in Table 1.

MIIGraph achieves superior results across both datasets. Specifically, on the ChFinAnn dataset, MIIGraph attains the highest F1-Score of 81.5%, with precision and recall rates of 85.1% and 78.1%, respectively. This surpasses the previous best-performing method, ReDEE, which achieved an F1-Score of 81.1%, precision of 84.0%, and recall of 79.9%.

Similarly, on the DuEE dataset, MIIGraph achieves an F1-Score of 75.8%, which is also the highest among all compared methods. Our precision and recall rates are 78.4% and 73.3%, respectively, significantly improving upon the prior best results obtained by ReDEE, which had an F1-Score of 74.4%, precision of 77.0%, and recall of 72.0%.

The notable improvement in performance highlights the effectiveness of our contrastive learning-based approach in capturing and utilizing the global theme information of documents for more accurate event extraction. This advancement underscores the importance of modeling comprehensive global theme information to enhance the precision and recall of DEE systems.

Table 1. Performance comparison on ChFinAnn and DuEE

Method	ChFinAnn			DuEE		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
DCFEE-O	69.7	57.8	63.2	51.9	49.6	50.7
DCFEE-M	60.1	61.3	60.7	37.3	48.6	42.2
GreedyDec	81.9	51.2	63.0	59.0	42.1	49.2
Doc2EDAG	81.1	77.0	79.0	67.1	51.3	58.1
GIT	82.4	77.6	79.9	70.3	46.0	55.6
PTPCG	83.7	75.4	79.4	63.6	53.4	58.1
ReDEE	84.0	79.9	81.1	77.0	72.0	74.4
ChatGPT	82.8	22.4	35.3	47.1	72.7	57.1
MIIGraph	85.1	78.1	81.5	78.4	73.3	75.8

4.3 Ablation Study

To validate the effectiveness of the components in MIIGraph, we conducted ablation experiments by evaluating four key designs. The F1-Score for the relevant tasks are presented in Tables 2 and 3.

Table 2. Ablation Experiments on ChFinAnn

Model	ChFinAnn		
	Precision	Recall	F1-Score
MIIGraph	85.1	78.1	81.5
-ThemeInformation	84.3 (-0.8)	77.0 (-1.1)	80.5 (-1.0)
-ContrastiveLearning	84.8 (-0.3)	76.7 (-1.4)	80.6 (-0.9)
-Craph	83.8 (-1.3)	75.7 (-2.4)	79.8 (-1.7)
-HGT	82.0 (-3.1)	78.0 (-0.1)	78.0 (-1.5)

- **Theme Information:** We investigated the effectiveness of modeling global theme information by removing this component from MIIGraph. As shown in

Table 3. Ablation Experiments on DuEE

Model	DuEE		
	Precision	Recall	F1-Score
MIIGraph	78.4	73.3	75.8
-ThemeInformation	76.6 (-1.8)	72.1 (-1.2)	74.2 (-1.6)
-ContrastiveLearning	77.9 (-0.5)	72.8 (-0.5)	75.2 (-0.6)
-Graph	78.3 (-0.1)	72.7 (-0.6)	75.4 (-0.4)
-HGT	78.1 (-0.3)	73.0 (-0.3)	75.5 (-0.3)

the results, the absence of global theme information modeling led to a noticeable drop in F1-Score for both datasets, indicating its crucial role in improving performance. Specifically, the F1-Score decreased by 1.0% on ChFinAnn and by 1.6% on DuEE.

- **Contrastive Learning:** To understand the impact of contrastive learning, we replaced it with a standard Transformer for modeling global theme information. The results demonstrate a decline in performance, with F1-Score dropping by 0.9% on ChFinAnn and by 0.6% on DuEE. This highlights the importance of contrastive learning in enhancing the representation of global theme information.
- **Graph:** We assessed the significance of using a heterogeneous graph to model the document by removing this aspect. The F1-Score dropped by 1.7% on ChFinAnn and by 0.4% on DuEE, indicating that the heterogeneous graph structure is beneficial for capturing complex document relationships.
- **HGT:** To explore the effectiveness of HGT, we substituted it with a multi-layer Graph Convolution Network(GCN) [10] for updating the heterogeneous graph. The results show that using GCN instead of HGT led to a decrease in performance, with F1-Score falling by 1.5% on ChFinAnn and by 0.3% on DuEE. This suggests that HGT is more effective in handling the intricacies of the heterogeneous graph.

These ablation experiments confirm that each component, “theme information modeling”, “contrastive learning”, “graph”, and “HGT” plays a vital role in the success of MIIGraph. The contributions of these components are evident in their collective enhancement of the model’s performance, as shown by the F1-Score.

5 Related Work

Previous research has primarily focused on sentence-level event extraction. For instance, DMCNN [2] enhanced event extraction performance by introducing multiple pooling mechanisms to capture different feature representations. A joint event extraction model based on RNNs [17] leveraged the sequence modeling capabilities to capture temporal dependencies of events and achieved improved

extraction performance. Viewed event extraction as a Machine Reading Comprehension (MRC) task transforms the problem into a reading comprehension challenge, enabling effective extraction of relevant event information [15]. An Attention-based Graph Information Aggregation model used graph structures to capture relationships between events and enhance the accuracy and robustness of event extraction [16]. The application of pre-trained language models for event extraction and generation significantly improved accuracy and fluency by capturing contextual information [22]. However, in real-life scenarios, the parameters of an event are often scattered across multiple sentences. Therefore, further research on document-level event extraction is necessary.

Due to the significant value of Document-level Event Extraction (DEE), considerable research efforts have been dedicated to this area. The DCFEE [20] framework initially performs Sentence-level Event Extraction (SEE) on central sentences and subsequently supplements event arguments from adjacent sentences. The Doc2EDAG [24] method generated event records using entity-based Directed Acyclic Graphs (DAGs), effectively improving the performance of DEE. The DE-PPN [21] adopted a pipeline paradigm to generate event records. The PTPCG [25] approach addressed time complexity, significantly improving the efficiency of DEE. GIT [19] designed a heterogeneous graph interaction network to capture global interaction information between different sentences and entity mentions. ReDEE [14] pioneered incorporating entity relationships into DEE, pursuing additional information to guide DEE processes. We build on this approach by exploring the influence of global theme information within documents on DEE.

6 Conclusion

In this paper, we propose the Multi-granularity Information Integration Graph (MIIGraph) that constructs a heterogeneous graph to capture the interaction of multi-granularity information such as entities, sentences, and the global theme of a document for document-level event extraction (DEE). Firstly, we utilize contrastive learning techniques to obtain a higher quality global theme representation of the document. Secondly, by constructing a heterogeneous graph based on global theme information, we effectively alleviate the challenges of extracting events at the DEE and improve the performance of this task.

In future work, we will continue to explore the impact of global theme information on document-level event extraction (DEE). Since large language models (LLMs) have a significant impact on natural language processing tasks, we will also explore using prompt-tuning or in-context learning of LLMs for document-level event extraction.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (No. 62206004, No. 62272001, No. 62106004), Hefei Key Common Technology Project (GJ2022GX15), and Xunfei Zhiyuan University Digital Transformation Innovation Research Project (2023ZY001).

References

1. Chakrabarti, S.: Deep knowledge graph representation learning for completion, alignment, and question answering. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2022, pp. 3451–3454. Association for Computing Machinery, New York (2022)
2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Zong, C., Strube, M. (eds.) Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, July 2015. Association for Computational Linguistics, pp. 167–176 (2015)
3. Collobert, R., Weston, J.: A unified architecture for natural language processing. In: Proceedings of the 25th International Conference on Machine Learning - ICML 2008, January 2008
4. Er-Rahmadi, B., Oncevay, A., Ji, Y., Pan, J.Z.: KATIE: a system for key attributes identification in product knowledge graph construction. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, pp. 3320–3324. Association for Computing Machinery, New York (2023)
5. Gao, J., Zhao, H., Yu, C., Xu, R.: Exploring the feasibility of chatGPT for event extraction. *CoRR*, abs/2303.03836 (2023)
6. Gao, T., Yao, X., Chen, D.: SimCSE: a simple contrastive learning of sentence embeddings. In: EMNLP 2021, January 2021
7. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020, April 2020
8. Huang, Y., Jia, W.: Exploring sentence community for document-level event extraction. In: Moens, M.-F., Huang, X., Specia, L., Yih, S.W. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, November 2021, pp. 340–351. Association for Computational Linguistics (2021)
9. Huang, Y., Jia, W.: Exploring sentence community for document-level event extraction. In: EMNLP 2021, January 2021
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016)
11. Li, Q., et al.: A survey on deep learning event extraction: approaches and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **35**, 6301–6321 (2022)
12. Li, S., Ji, H., Han, J.: Document-level event argument extraction by conditional generation. In: Toutanova, K., et al. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 894–908, Online. Association for Computational Linguistics, June 2021
13. Li, X., et al.: DuEE: a large-scale dataset for Chinese event extraction in real-world scenarios. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020, Part II. LNCS (LNAI), vol. 12431, pp. 534–545. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_44
14. Liang, Y., Jiang, Z., Yin, D., Ren, B.: RAAT: relation-augmented attention transformer for relation modeling in document-level event extraction. In: Carpuat, M., de Marneffe, M.-C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics:

- Human Language Technologies, Seattle, United States, July 2022, pp. 4985–4997. Association for Computational Linguistics (2022)
- 15. Liu, J., Chen, Y., Liu, K., Bi, W., Liu, X.: Event extraction as machine reading comprehension. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, November 2020, pp. 1641–1651. Association for Computational Linguistics (2020)
 - 16. Liu, X., Luo, Z., Huang, H.: Jointly multiple events extraction via attention-based graph information aggregation. In: EMNLP 2018, January 2018
 - 17. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: NAACL 2016, January 2016
 - 18. Vaswani, A., et al.: Attention is all you need. In: NeurIPS 2017, June 2017
 - 19. Xu, R., Liu, T., Li, L., Chang, B.: Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In: ACL 2021, January 2021
 - 20. Yang, H., Chen, Y., Liu, K., Xiao, Y., Zhao, J.: DCFEE: a document-level Chinese financial event extraction system based on automatically labeled training data. In: Liu, F., Solorio, T. (eds.) Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, July 2018, pp. 50–55. Association for Computational Linguistics (2018)
 - 21. Yang, H., Sui, D., Chen, Y., Liu, K., Zhao, J., Wang, T.: Document-level event extraction via parallel prediction networks. In: ACL 2021, January 2021
 - 22. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: ACL 2019, January 2019
 - 23. Zhang, Z., Wang, B.: Prompt learning for news recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, pp. 227–237. Association for Computing Machinery, New York (2023)
 - 24. Zheng, S., Cao, W., Xu, W., Bian, J.: Doc2EDAG: an end-to-end document-level framework for Chinese financial event extraction. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, November 2019, pp. 337–346. Association for Computational Linguistics (2019)
 - 25. Zhu, T., et al.: Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. In: IJCAI 2022, July 2022



Multi-granularity Neural Networks for Document-Level Relation Extraction

Xiye Chen¹  and Peng Wang² 

¹ Nanjing University of Finance and Economics, Nanjing 210023, China

² Southeast University, Nanjing 210096, Jiangsu, China

pwang@seu.edu.cn

Abstract. Document-level relation extraction aims to extract relations from multiple sentences in a document. However, it remains challenging to obtain rich semantic information across multiple sentences for relation prediction. In this paper, a multi-granularity relation extraction (*MGRE*) neural network is proposed, which integrates multiple granularity semantic features (i.e., entity level, sentence level and document level), to capture the semantic interactions among entities and sentences in a document. For entities, the shortest dependency path is utilized to obtain head-to-tail entity representations, which is further used for acquiring the relation between entity pairs through a translation strategy. Then, at the sentence level, a convolutional neural network is used to extract semantic features for each sentence. While for the documents, an attention mechanism is adopted to fuse multiple sentence-level feature vectors into document-level semantic features. Finally, entity representation and document representation are combined into a holistic representation for relation prediction. Extensive experiments are conducted on the DocRED dataset against state-of-the-art methods, and the comparative results demonstrate the superiority of *MGRE* on document-level relation extraction.

Keywords: Document-level relation extraction · Syntactic dependency analysis · Knowledge graph

1 Introduction

Relation extraction (RE) aims to extract relational facts between entities from unstructured text, which is one of the most fundamental tasks in information extraction and semantic analysis. Most previous methods focus on extracting relations from a single sentence [15, 19, 31]. However, in real-world applications, valuable relations such as the relations between diagnosis and treatment, are usually obtained across multiple sentences. Therefore, document-level relation extraction is pivotal for understanding documents.

Document-level RE often requires reasoning on multiple sentences to extract corresponding entities and relations, which still suffers from some deficiencies.

[1] Le Reculet is the second highest summit in the Jura Mountains. [2] It is located in the Ain department in France. [3] Its altitude is 1718 metres. [4] It is situated a few kilometres south of the Cret de la Neige on the territory of the town of Thoiry. [5] A cross was erected on the summit by the inhabitants of Thoiry. [6] The summit has views of the Pays de Gex, Geneva, Lake Geneva, the Alps, Mont Blanc, the Matterhorn, and on clear days the Chaine des Puys. [7] Le Reculet was sometimes designated as the highest point of the Jura, until the elevation of the Cret de la Neige was revised upwards to 1720 m instead of 1717.6 m.

Subject	Object	Relation	Supporting Evidence
Le Reculet	France	country	1,2
Le Reculet	Jura Mountains	part_of	1,7
Ain	France	country	2

Fig. 1. An example of document-level relation extraction.

First, the entities corresponding to a relation may appear in different sentences, as a result, it is difficult to determine the relations between entities based on only one sentence. Second, documents contain multiple granularities information over entity-level, sentence-level and document-level, in that way, how to obtain and integrate features of all granularities is a nontrivial process for relation prediction. Additionally, since that entities in different sentences have dependencies, it is necessary to obtain and exploit such dependencies during RE. As shown in Fig. 1, since the head and tail entities appear in the same sentence, it is easy to identify the intra-sentential relational triples (*Le Reculet, part_of, Jura Mountains*) and (*Ain, country, France*). However, for the entities *Le Reculet* and *France*, which are in sentence 1 and sentence 2 separately, it is more difficult to obtain the triple (*Le Reculet, country, France*), because entity references do not appear in the same sentence and have long-distance dependencies. Moreover, the identification of inter-entity relations needs to deal with the problem of referents, which also requires the assistance of syntactic dependencies.

Fortunately, some document-level relation extraction methods have explored multi-granularity information and entity dependencies. Gupta et al. [6] enriched the semantic features of sentences by constructing the shortest dependency paths and augmented dependency paths for sentence features encoding through syntactic dependency parsing. However, their method only considers the path sequence of entities, which overlooks the correlation between entity pairs. Kim et al. [7] proposed a global document-level relation extraction model using knowledge graph embedding. Their model uses vector representations of local relations to construct knowledge graphs to represent input documents, and then uses the knowledge graphs to predict global-level relations from documents or large amounts of text. Wang et al. [25] proposed a model for encoding document information based on BERT [3], which classifies relations in two stages: (1) determining whether there is a relation between entity pairs, and (2) determining the corresponding relation. Nevertheless, this method only extracts text features, which is not sufficient to fully represent the connection between

entities and relations in a document. Tang et al. [23] proposed a hierarchical inference network by focusing on feature information at different granularities in documents, which makes full use of the information at entity-level, sentence-level and document-level, and can effectively gather inference information from different granularities. However, their work only obtains entity-level feature representations from the text sequence level using BiLSTM [20], which misses some association between entities and other words in the text.

To obtain the intra-sentence dependencies of entities and fully integrate the semantic information of multiple granularities, in this paper, a multi-granularity relation extraction neural network (*MGRE*) is proposed, which can better collect and synthesize the inter-sentence information. *MGRE* first gets the entity representation via the entity-level network, which uses the dependency path to capture the semantic information of two entities and adopts the translation strategy to fuse the entity pair information into the final entity representation. Then, it uses sentence-level convolutional neural networks to extract semantic information for each sentence. Furthermore, it applies an attention mechanism to the multiple sentence-level representations to generate the document representation in the document-level network. Finally, *MGRE* merges the entity representation and document representation into final representation for relation prediction.

We evaluate *MGRE* on DocRED [30], a public available document-level dataset, where it outperforms other competitive baselines. The results show that the fusion of multi-granularity information could effectively improve the performance of RE. We perform detailed analysis and ablation experiments to highlight the significance of using the shortest dependency path and using the translation strategy of TransE [1] to capture and fuse the semantic entity pair information into a unified entity level. We also conduct case studies to compare *MGRE* with other models, all the results demonstrate the advantage of combining multi-granularity information in improving document-level RE.

2 Related Work

Early RE research focuses on predicting relations at the sentence level. Socher et al. [21] used an RNN to address the RE problem, effectively incorporating syntactic structure information. Similarly, CNN is applied for RE [19, 31]. Wang et al. [26] used a CNN model with a two-layer attention mechanism, emphasizing sentence parts contributing more to relation labels, significantly improving classification. Miwa and Bansal [15] used bidirectional LSTM and tree LSTM to model entities and sentences. Lin et al. [9] proposed a neural network with a sentence-level attention mechanism, assigning weights to each sentence of an entity pair based on a specific relation. These methods effectively address the sentence-level RE problem.

Recent research focuses on document-level RE due to relations spanning multiple sentences. Gu et al. [5] used a maximum entropy model and CNN for inter-sentence and intra-sentence RE, respectively, aggregating results for document-level RE. Verga et al. [24] proposed the bi-radial relational attention network

for simultaneous prediction of relations among all entity pairs in a document. Nguyen and Verspoor [17] integrated character-based word representations into a CNN-based RE model. Wang et al. [25] developed a BERT-based document information encoding model [3]. Tang et al. [23] introduced a hierarchical inference network (HIN) using rich information at entity, sentence, and document levels. For non-local interactions, Xu et al. [28] argued for unique dependencies between entity pairs in relational triples, incorporating them into self-attention mechanisms and the encoding phase.

Recent document-level RE relies on document graphs. Christopoulou et al. [2] utilized diverse nodes and edges for graph creation, with inference mechanisms employing internal multi-instance learning for intra-sentence and inter-sentence relations. Kim et al. [7] employed knowledge graphs to predict global-level relations from documents or extensive text. Nan et al. [16] enhanced relational inference between sentences by automatically introducing potential document-level graphs. Zeng et al. [32] proposed graph aggregation and inferential networks for document-level RE. Zhou et al. [34] introduced a global context-enhanced graph convolutional network (GCGCN) for capturing rich global contextual information of entities in a document.

3 Methodology

3.1 Problem Statement

The document-level RE is to extract all possible relations among all entities from the document. Let $D = \{S_i\}_{i=1}^{n_s}$ and $V = \{E_i\}_{i=1}^{n_e}$ denote the annotated document and corresponding entity set, in which n_s denotes the number of sentences and n_e denotes the number of entities. For each sentence in the document D , the i -th sentence containing n_w^i words is denoted by $S_i = \{w_j\}_{j=1}^{n_w^i}$, and the i -th entity with n_m^i mentions is denoted by $E_i = \{m_j\}_{j=1}^{n_m^i}$. Our goal is to train a relation extraction model M to predict all relations $R' \in R = \{r_i\}_{i=1}^{n_r}$ between each entity pair, i.e., $M(D) \rightarrow R' \in R$, where n_r denotes the number of relations.

3.2 Model Overview

To address the document-level RE issues mentioned above, especially, to capture the semantic interaction between entities and sentences, this paper proposes MGRE to fully integrate multi-granularity semantic information from entity-level to document-level. The framework of MGRE is shown in Fig. 2.

MGRE mainly consists of four components: (1) The entity-level network, which consumes the input entity pair and their corresponding sentences to get the overall semantic representation of the entity pair through dependency analysis and translation strategy; (2) The sentence-level network, which uses convolutional neural networks to extract semantic information from each input sentence; (3) The document-level network, which applies an attention mechanism to fuse

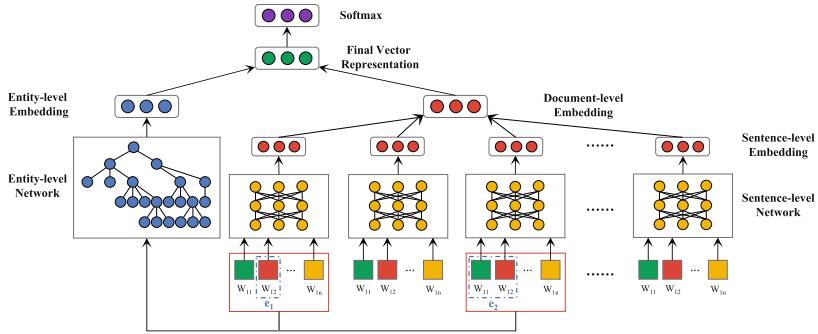


Fig. 2. The framework of MGRE

sentence representations obtained by sentence-level networks to generate the document representation; (4) And the final classification layer, which combines the entity representation and document representation to final representation and projects it to the probabilities for each relation type.

3.3 Entity-Level Network

In the entity-level network, we try to capture the entity pair information using the shortest dependency path, then extract the semantic feature representation of the entity through LSTM, and fuse the information of two entities into the final feature representation through the translation strategy. The structure of entity-level network is shown in Fig. 3.

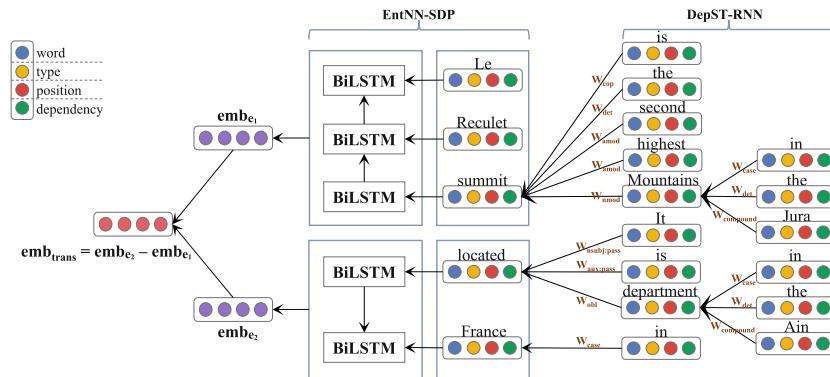


Fig. 3. The structure of entity-level network.

Modeling the Shortest Dependency Path. The shortest dependency path (SDP) is the shortest path from two entities to their common ancestor node in

the syntactic dependency tree. In relation extraction, SDP can make the model to capture more on relevant information and ignore irrelevant information.

We adopt Stanford parser to obtain the syntactic dependency tree corresponding to the given sentence. For entity pairs in the same sentence, the syntactic dependency tree can be directly obtained. If two entities are in different sentences, we first obtain two syntactic dependency trees respectively, and then set a virtual common node to connect the two syntactic dependency trees. Taking sentences 1 and 2 in Fig. 1 as an example, these two sentences work together to get a relation triple (*Le Reculet*, *country*, *France*). Figure 4 shows that the dependency trees of the two adjacent sentences are connected by a virtual common node. The SDP of entity pair *Le Reculet* and *France* is highlighted by red color.

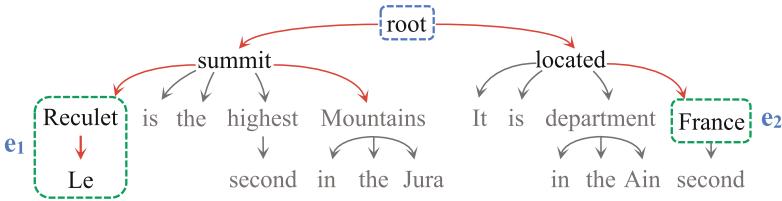


Fig. 4. The syntactic dependency tree for two sentences.

In order to better represent the word nodes on the SDP, we model dependency subtrees assuming that each word node w can be seen as the word itself and its children on the dependency subtree.

Modeling Dependency Subtrees. The goal of modeling dependency subtrees is to learn a good representation for the words on the shortest path. We assume that each word w can be interpreted by itself and its children on the dependency subtree. Inspired by dependency-based neural network [11], we propose the DepST-RNN to learn the representation for multiple sentences.

For each word w on the dependency tree, its word embedding vector $\mathbf{x}_w \in \mathbb{R}^{n_1}$, entity type vector $\mathbf{q}_w \in \mathbb{R}^{n_2}$, position embedding vector $\mathbf{p}_w \in \mathbb{R}^{n_3}$ and subtree representation $\mathbf{c}_w \in \mathbb{R}^{n_4}$ are concatenated to form its final representation $\mathbf{t}_w \in \mathbb{R}^{n_1+n_2+n_3+n_4}$. We adopt GloVe and BERT to get word embeddings. For entity type vector representation, we map each entity type to a separate vector representation. Similar to [31], we use position embeddings specified by entity pairs which is defined as the combination of the relative distances from the current word to head or tail entities. Taking sentence [2] in Fig. 1 for example, the relative distance between *department* to head entity *Ain* is 1 and tail entity *France* is -2. Then we adopt recursive neural network (RNN) to construct the subtree embedding \mathbf{c}_w from the leaf node to the root node. The detail structure of DepST-RNN can be seen in Fig. 3. Each word in the dependency tree is connected by the dependency relation r . For each relation r , a learnable transfer

matrix $\mathbf{W}_r \in \mathbb{R}^{n_4 \times (n_1+n_2+n_3+n_4)}$ is set. The subtree embedding is calculated as:

$$\mathbf{c}_w = f\left(\sum_{u \in \text{children}(w)} \mathbf{W}_r(w, u) \cdot \mathbf{t}_u + \mathbf{b}\right) \quad (1)$$

where $\mathbf{t}_u = [\mathbf{x}_u, \mathbf{q}_u, \mathbf{p}_u, \mathbf{c}_u]$, $r(w, u)$ denotes the dependency between word w and its child word u . $\mathbf{b} \in \mathbb{R}^{n_4}$ is a bias.

Entity Representation. In RE, the entity representation is particularly important. To fully integrate the structural information of the shortest dependency path and dependency subtree, we propose the entity neural network based on the shortest dependency path (EntNN-SDP) to better capture the entity information. As shown in Fig. 3, LSTM is used to extract semantic features along the path where the entity e_1 and e_2 are located. Basically, a LSTM unit is composed of three multiplicative gates which control the proportions of information to forget and to pass on to the next time step. The LSTM unit is:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{U}_i \cdot \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{U}_f \cdot \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{U}_o \cdot \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (4)$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_g \cdot \mathbf{x}_t + \mathbf{U}_g \cdot \mathbf{h}_{t-1} + \mathbf{b}_g), \quad (5)$$

$$\mathbf{c}_t = \mathbf{i}_t \circ \mathbf{g}_t + \mathbf{f}_t \circ \mathbf{c}_{t-1}, \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \quad (7)$$

where σ is the sigmoid function, and \circ is the element-wise product. \mathbf{i} , \mathbf{f} , \mathbf{o} , \mathbf{g} and \mathbf{c} are the input gate, forget gate, output gate, candidate memory cell and memory cell at time t , respectively. \mathbf{x}_t is the input vector at time t , and \mathbf{h}_t is the hidden state (also called output) vector storing all the useful information at (and before) time t . The $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{W}_g$ and $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_o, \mathbf{U}_g$ are weight matrices, and $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_g$ are bias vectors.

We take the final hidden vector \mathbf{h}_t of LSTM as the representation of an entity. The vector representation of entity e_1 is:

$$\mathbf{emb}_{e_1} = \mathbf{h}_t^{e_1} \quad (8)$$

Similarly, the vector representation of e_2 is:

$$\mathbf{emb}_{e_2} = \mathbf{h}_t^{e_2} \quad (9)$$

Then we obtain an overall semantic representation of e_1 and e_2 . Here, we propose two strategies: the concatenation strategy and the translation strategy.

The Concatenation Strategy. To calculate multiple feature vectors, we concatenate the two vectors to obtain an overall vector representation.

$$\mathbf{emb}_{cat} = [\mathbf{emb}_{e_1}, \mathbf{emb}_{e_2}] \quad (10)$$

The Translation Strategy. The concatenation strategy simply concatenates two entity vectors and cannot obtain the interaction information between them.

Inspired by the widely used TransE model [1], which regards the embedding of a relation r as the difference between two entity embeddings, we will also adopt this method for the overall representation of entities.

$$\mathbf{emb}_{trans} = \mathbf{emb}_{e_2} - \mathbf{emb}_{e_1} \quad (11)$$

where \mathbf{emb}_{trans} is the final entity representation.

3.4 Sentence-Level Network

We use convolutional neural network (CNN) to construct sentence-level semantic information, and the structure of the sentence-level network is shown in Fig. 5.

Word Representation. Given a sentence S consisting of n words $S = \{w_1, w_2, \dots, w_n\}$, where w_i denotes the i -th word in the sentence, and the corresponding representation is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$, where each \mathbf{x}_i is a concatenation of word vector, entity type vector and position vector and $\mathbf{x}_i \in \mathbb{R}^d$. The word representation is similar to the input in entity-level network, but does not have the dependency embedding.

Convolution. For word sequences of textual type, word representation can capture contextual information through combinations of vectors in a window. However, it only produces local features around each word. In relation extraction, an input sentence that is marked with the target entity pair corresponds to one or more relation types, rather than predicting a label for each word. Therefore, it is necessary to use all local features to predict relations globally. The convolution is a natural method to merge all of the features.

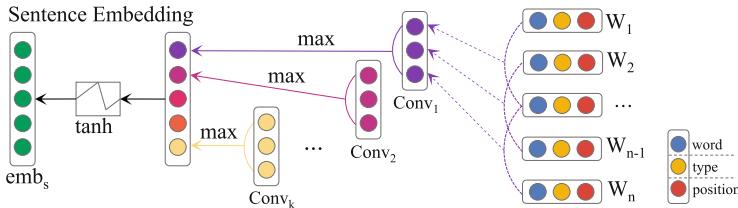


Fig. 5. The structure of sentence-level network.

Convolution is an operation between a convolution matrix \mathbf{w} and the vector sequence \mathbf{X} of the input, and the convolution matrix \mathbf{w} is a filter of the convolution network, each of which can extract different features from the input. In the example shown in Fig. 5, we assume that the length of the filter l is 3, $\mathbf{w} \in \mathbb{R}^{l \times d}$. Note that $\mathbf{x}_{i:j}$ is the sequence from x_i to x_j , and the convolution operation is to take the dot product of \mathbf{w} with the sequence of each l length in \mathbf{X} , so as to obtain a feature sequence $c \in \mathbb{R}^{n-l+1}$.

$$c_j = \mathbf{w} \cdot \mathbf{x}_{j:j+l-1} \quad (12)$$

To capture features from different dimensions, multiple filters are used in the convolution. This paper uses k different convolution kernels $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ to convolve \mathbf{X} with those kernels.

$$c_{ij} = \mathbf{w}_i \cdot \mathbf{x}_{j:j+l-1} \quad (13)$$

After convolution operation, we can get the feature matrix $\mathbf{C} = \{c_1, c_2, \dots, c_k\} \in \mathbb{R}^{k \times (n-l+1)}$.

Max Pooling. The size of the output matrix \mathbf{C} of the convolution layer depends on the length of the input sentence. In order to apply the output to the subsequent network layer, the features obtained by the convolution layer must be combined to obtain a fixed-sized vector for the input sentence. The max pooling operation can naturally address variable sentence lengths and capture the most significant features (with the highest values) in each feature map. We perform a max pooling operation over time on \mathbf{C} .

$$m_i = \max \{\mathbf{C}(i, \cdot)\}, 1 \leq i \leq k \quad (14)$$

where $\mathbf{C}(i, \cdot)$ denotes the i -th row of matrix \mathbf{C} . Finally, we obtain the feature vector $\mathbf{m} = \{m_1, m_2, \dots, m_k\}$, and the dimension of \mathbf{m} is fixed and no longer related to the sentence length.

Sentence Representation. For the feature vector obtained from the pooling layer, in order to obtain the whole sentence representation emb_S , we design a nonlinear layer and selected hyperbolic tangent function \tanh as the activation function. A useful property of \tanh is that its derivative can be represented by the function value itself.

$$\frac{d}{dx} \tanh x = 1 - \tanh^2 x \quad (15)$$

It has the advantage of making it easy to compute the gradient in the back-propagation training procedure. Formally, the nonlinear transformation is:

$$\mathbf{emb}_S = \tanh \mathbf{m} \quad (16)$$

In this paper, we adopt GloVe and BERT to get those word vectors. For entity type vector representation, we map each entity type to a separate vector representation. We employ the position embedding method proposed by Zeng et al. [31] to get our position vector.

After \mathbf{X} is constructed, we use k different convolution kernels $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\}$ to convolve \mathbf{X} with those kernels. Then the max pooling is used to fuse those word-level representations into a sentence-level vector, with each pooling result makes up one dimension of the sentence-level vector. We use the max pooling to address the problem of different sequence length. Eventually a nonlinear transformation (e.g. \tanh) is applied to transform the sentence-level vector into sentence representations.

3.5 Document-Level Network

We adopt the self-attention proposed by Lin et al. [10] to learn document-level information, which is vital for relation classification from sentence-level representations. Specifically, we compute an attention score α_i for each sentence S_i as follow.

$$\alpha_i = \mathbf{v}_a \cdot \tanh(\mathbf{W}_a \cdot \mathbf{emb}_{S_i}) \quad (17)$$

where \mathbf{W}_a and \mathbf{v}_a are parameters in attention mechanism. The attention scores are then normalized to fuse sentence vectors into document embeddings.

$$\mathbf{emb}_D = \sum_{1 \leq i \leq M} \left\{ \frac{\exp(\alpha_i)}{\sum_{1 \leq j \leq M} \exp(\alpha_j)} \cdot \mathbf{emb}_{S_i} \right\} \quad (18)$$

where M is the number of sentences in the document.

3.6 Relation Classification

To better exploit the semantic information at different granularities, we integrate the entity-level vector and the document-level vector as the final representation. We perform linear transform on both the entity-level representation and the document-level representation following the equations as follows.

$$\mathbf{o}_E = \mathbf{W}_E \cdot \mathbf{emb}_E + \mathbf{b}_E \quad (19)$$

$$\mathbf{o}_D = \mathbf{W}_D \cdot \mathbf{emb}_D + \mathbf{b}_D \quad (20)$$

where \mathbf{W}_E and \mathbf{W}_D are weight matrices, \mathbf{b}_E and \mathbf{b}_D are bias vectors, and $|\mathbf{o}_E| = |\mathbf{o}_D| = N_r$ denote the number of relation types.

We take the weighted sum of \mathbf{o}_E and \mathbf{o}_D to get the overall score vector \mathbf{o} .

$$\mathbf{o} = \alpha \circ \mathbf{o}_E + (1 - \alpha) \circ \mathbf{o}_D \quad (21)$$

3.7 Training Objectives

In this paper all training data are represented as $D = \{(S_1, r_1), (S_2, r_2), \dots, (S_M, r_M)\}$, where each S_i is a set of sentences and r_i are corresponding relations. Given a document D , since each candidate entity pair may contain multiple relations, we consider the multi-classification problem into multiple binary classification problems. For each input sample, the network outputs a score vector \mathbf{o} , where \mathbf{o}_i represents the score of the i -th relation. To obtain the conditional probability $p(r_i|D, e_1, e_2)$ for relation r_i , a sigmoid operation is applied to relation types:

$$p(r_i|D, e_1, e_2) = \frac{1}{1 + e^{-\mathbf{o}_i}}, i \in [1, N_r] \quad (22)$$

where N_r is the number of relation types.

In training, we take the binary cross entropy as loss function:

$$\begin{aligned} loss = & - \sum_{D \in D_T} \sum_{e_1 \neq e_2} \sum_{r_i \in \mathcal{R}} (y_i \log p(r_i | D, e_1, e_2) \\ & + (1 - y_i) \log(1 - p(r_i | D, e_1, e_2))) \end{aligned} \quad (23)$$

where D_T denotes the whole corpus, \mathcal{R} is a pre-defined relation type set, and $y_i \in \{0, 1\}$ denotes the true label of relation r_i .

4 Experiments

4.1 Dataset and Evaluation Metrics

We evaluate MGRE on public document-level RE dataset DocRED [30]. DocRED is a large-scale manually annotated dataset for document-level RE, and it is constructed from Wikipedia articles. There are 46.4% relation instances associated with more than one supporting sentence, about 7% of entity pairs have more than one relation, and 40.7% of relational facts can only be extracted from multiple sentences [30]. DocRED includes 6 entity types: person, location, organization, time, number and other. To this end, DocRED requires rich reasoning skills for synthesizing all information of the document, like logical reasoning, coreference reasoning and common-sense reasoning, which is still challenging for existing models. The statistics of DocRED are shown in Table 1. We use F1-score and Ignore F1-score as evaluation metrics. There are some related triples in the training set and testing set, which may influence the evaluation performance, so we record the performance after filtering out these triples (i.e., Ign F1).

Table 1. Statistics of dataset DocRED.

Setting	# Doc.	# Rel.	# Inst.	# Fact
Train	3,053	96	38,269	34,715
Dev	1,000	96	12,332	11,790
Test	1,000	96	12,842	12,101

4.2 Baseline

We compare our proposed MGRE model against some state-of-the-art document-level RE models: Yao et al. [30] proposed **CNN**, **LSTM**, **BiLSTM** to encode a document into a sequence of hidden state vectors, and **Context-aware** that adds the attention mechanism to the **LSTM**. **GREG** is a global RE model proposed by Kim et al. [7], which uses a knowledge graph to embed document-level input. **BERT-Two-Step** is a BERT-based document encoding model proposed by Wang et al. [25], which first determines whether there is a relation between entity pairs, and then predicts the corresponding relation. **HIN** is a hierarchical

reasoning network proposed by Tang et al. [23]. It utilizes the rich information at the entity-level, sentence-level, and document-level, which can effectively gather reasoning information from different granularities. In order to improve inter-sentence reasoning, **GEDA** proposed by Li et al. [8] uses graph-enhanced dual attention networks to describe complex interactions between sentences and potential relation instances. **GCGCN** proposed by Zhou et al. [34] takes entities as nodes and the context of entity pairs as edges between nodes, and then captures the global context of entities in the document. Nan et al. [16] proposed **LSR** to enhance the reasoning of the relation between sentences by automatically introducing potential document-level graphs. Xu et al. [28] proposed **SSAN** to combine and incorporate various entity structural dependencies defined under a unified framework into the standard self-attention mechanism and the entire coding stage. **GAIN** proposed by Zeng et al. [32] constructs two graphs, a heterogeneous mention level graph and an entity level graph, on which a novel path reasoning mechanism based to infer the relations between entities. **AA** proposed by Lu et al. [13] explicitly and jointly leverages coreference and anaphora information in the document and employs an attention-based graph convolutional neural network to dynamically learn the structure of the graph. **DocRE-II** is proposed by Zhang et al. [33] as a novel document-level RE model with iterative inference. **DREEAM** is proposed by Ma et al. [14] as a memory-efficient approach that adopts evidence information as the supervisory signal, which guides the attention modules of the DocRE system to assign high weights to evidence. **Eider** is an evidence-enhanced framework for DocRE proposed by Xie et al. [27], which efficiently extracts and fuses evidence with a lightweight extraction model jointly trained with the main RE model. **SIEF** framework is proposed by Xu et al. [29], where a sentence importance score and a sentence focusing loss are designed to encourage DocRE models to focus on evidence sentences instead of the entire document. **KD-DocRE** is a semi-supervised framework for DocRE proposed by Tan et al. [22] that incorporates an axial attention module for learning the interdependency among entity-pairs.

4.3 Implementation Details

MGRE is optimized with AdamW [12] using learning rate as $1e-5$ with a linear warmup [4] for the first 1000 steps followed by a weight decay rate as 0.001. For the word vector setting, this paper adopts two kinds of settings: (1) We set the dimension of GloVe [18] vectors to 100 and set the batch size to 16. (2) We transform the pretrained cased BERT-base [3] embeddings into 100 dimensions with a fully connected layer and set the batch size to 2 due to the limitation of hardware. For the entity type embedding, we first number each entity type, and then map them to fixed-dimensional dense vectors to represent the corresponding entity type information. For the vector dimension, we follow the setting of DocRED [30].

Table 2. Performances of different models on DocRED. Models above the first double line use GloVe embedding, and other models use BERT embedding. Results with * are reported in original papers.

Model	Dev		Test	
	Ign F1 (%)	F1 (%)	Ign F1 (%)	F1 (%)
CNN* (Yao et al., 2019)	41.58	43.45	40.33	42.26
LSTM* (Yao et al., 2019)	48.44	50.68	47.71	50.07
BiLSTM* (Yao et al., 2019)	48.84	50.94	48.78	51.06
Context-aware* (Yao et al., 2019)	48.94	51.09	48.40	50.70
GREG* (Kim et al., 2020)	—	—	—	52.88
HIN-GloVe* (Tang et al., 2020)	51.06	52.95	51.15	53.30
GCGCN-GloVe* (Zhou et al., 2020)	51.14	53.05	50.87	53.13
LSR-GloVe* (Nan et al., 2020)	48.82	55.17	52.15	54.18
GAIN-GloVe* (Zeng et al., 2020)	53.05	55.29	52.66	55.08
MGRE-GloVe	55.36	56.96	54.89	56.25
BERT-Two-Step* (Wang et al., 2019)	—	54.42	—	53.92
HIN-BERT* (Tang et al., 2020)	54.29	56.31	53.70	55.60
GEDA-BERT* (Li et al., 2020)	56.16	54.52	53.71	55.74
GCGCN-BERT* (Zhou et al., 2020)	55.43	57.35	54.53	56.67
LSR-BERT* (Nan et al., 2020)	52.43	59.00	56.97	59.05
SSAN-BERT* (Xu et al., 2020)	56.68	58.95	56.06	58.41
GAIN-BERT* (Zeng et al., 2020)	59.14	61.22	59.00	61.24
AA-BERT* (Lu et al., 2023)	61.31	63.38	60.84	63.10
DocRE-II-BERT* (Zhang et al., 2022)	60.75	62.74	60.68	62.65
DREEAM-BERT* (Ma et al., 2023)	60.51	62.55	60.03	62.49
Eider-BERT* (Xie et al., 2022)	60.51	62.48	60.42	62.47
SIEF-BERT* (Xu et al., 2022)	59.82	62.24	59.87	62.29
KD-DocRE-BERT* (Tan et al., 2022)	60.08	62.03	60.04	62.08
MGRE-BERT	61.75	63.88	61.24	63.27

4.4 Main Results

Table 2 reports the results of our proposed MGRE model against other baseline methods on the DocRED dataset. We divide these models into three groups: (1) GloVe-based models without hierarchical inference, including CNN, LSTM, BiLSTM and Context-aware; (2) GloVe-based models with hierarchical inference; (3) BERT-based models with hierarchical inference.

From results in Table 2, we can see that, among the Glove-based models, MGRE outperforms all other approaches. Concretely, Ign F1 and F1 are improved by at least 2.23% and 1.17%, respectively. Different from HIN model, which obtains the multi-granularity semantic information by sequence feature

processing based only on BiLSTM, in order to fully capture the dependency between words, MGRE further adds syntactic dependency information to entity features, and hence achieves better results.

Among BERT-based models, MGRE also outperforms other models. On the test set, Ign F1 and F1 are improved by at least 0.44% and 0.50%, respectively. Note that the semantic vectors obtained by BERT have been processed by multi-layer Transformer, resulting in the original syntactic dependency structure of the text that cannot be adequately preserved. It in turn could limit the performance of entity-level information processing network, and hence limits entity-level semantic information cannot be fully constructed. Therefore, the performance of MGRE could be improved by further exploring the entity-level semantic information in further studies.

Different from the early dependency analysis model, we first innovatively propose to construct entity information representation by using the shortest dependency path, and then fuse entity pair information into a unified entity level semantic representation by using the translation strategy of TransE, which further enriches the representation of entity-level semantic information. For document-level RE, it is necessary to deal with multiple sentences and entity pairs, while the information of sentences is constant, so how to highlight the semantic information of each entity pair in sentences is particularly important. Specially, MGRE greatly improves the representation of entity pairs and experimental results show that it helps to improve the performance of RE.

Finally, we argue that MGRE is not a complicated and unusable model: (1) First, compared with sentence-level RE, the semantic information is richer and more complex in document-level RE. To fully utilize the semantic information at all levels to better capture the semantic information among entities, while the architecture of MGRE seems a slightly complex, but it is effective for document-level RE. Therefore, the little complexity of MGRE is worth. (2) We optimize the model training by saving some information in advance to reduce the complexity of space and time. (3) Moreover, in the sentence-level network, the processing of each sentence is accelerated parallelly, which can greatly improve the training speed. (4) Actually, although MGRE is more complicated than GAIN, there is no significant difference in their training time.

4.5 Ablation Study

To further analyze the effect of different components, we conduct an ablation study on DocRED dev set (see Table 3). -Pos represents removing position embedding; -Dep represents removing dependency embedding; -Trans+Cat represents using concatenation strategy instead of translation strategy; -EntSDP denotes removing the entity information construction network; -SentNet stands for removing the sentence level network. As shown in Table 3, from these ablations, we can observe some interesting facts.

First, -Type, -Pos, and -Dep are indispensable features that bring 1.78%, 2.23% and 4.54% improvements in F1 score to ultimate performance, respectively, which indicates that they all enhance the representation ability of

Table 3. Ablation study of MGRE-BERT on DocRED dev set.

Setting	Ign F1 (%)	F1 (%)
MGRE-BERT	61.75	63.88
– Type	54.90	62.10
– Pos	54.49	61.65
– Dep	52.28	59.34
– EntSDP	52.67	59.60
– Trans + Cat	52.83	59.87
– SentNet	50.79	57.66

MGRE in varying degrees. Second, the entity information construction network (EntSDP) is crucial, because the F1 drops markedly by 4.28% if it is removed. It can be interpreted that dependency information can better capture the correlation between words in sentences, enriching the feature representation of words and providing great support for subsequent entity information construction. Third, when we use concatenation strategy (Cat) instead of translation strategy (Trans), F1 score drops by 4.01%, which suggests that translation strategy greatly improves the fusion representation of entity pair information. Finally, removing the sentence level network hurts the final result by 6.22%, which shows that the sentence level network can better extract the semantic features of different sentences.

4.6 Case Study

We analyze a document with nine sentences from DocRED for case study to provide a deeper understanding of MGRE and reflect the superiority of our proposed method. As shown in Fig. 6, the ground truth in the given case contains five relation triples. Note that some relations are missed in last 5 sentences, namely, it is an annotation missing example.

In the eight triples predicted by BiLSTM, only two relation triples are correctly identified, which indicates that BiLSTM cannot fully capture the document-level long-distance dependency well. Note that the two correct triples are relatively simple, and could be obtained by intra-sentence relation instead of cross sentence reasoning. Meanwhile, it leads to a lot of redundant triples due to too much interference information.

GCGCN-BERT predicts four correct relation triples, but still misses one correct triple and produces three wrong triples, which greatly degrades the accuracy of the result and shows that it lacks the ability to integrate all granularity semantic information comprehensively. For the predicted triples that are not in the ground truth, GCGCN-BERT actually gives predictions that are close to the correct answer. For example, GCGCN predicts the triple (*Excite*, *developer*, *Nintendo*) while the correct triple is (*Excite*, *publisher*, *Nintendo*). The relation

[1] **Excitebots: Trick Racing**, known in Japan as , is a racing video game published by **Nintendo** for the **Wii** video game console. [2] It was developed by **Monster Games**, is the fourth main game in the **Excite** series and is the sequel to **Excite Truck**. [3] **Excitebots** was unveiled in a release list from **Nintendo of America** on **February 26, 2009**. [4] It was released on **April 20, 2009** in **North America**. [5] **Excitebots** features animal-themed robot vehicles and short minigames during racing, such as pie throwing, bowling and soccer. [6] The game could be bought packaged with or without the **Wii Wheel**. [7] Despite a positive reception from critics, the game was never available in stores outside **North America**. [8] **Nintendo Australia**'s Managing Director, **Rose Lappin** has said that **Excitebots** will not be seeing an **Australian** release due to lack of interest. [9] However, **Japanese Club Nintendo** members were able to exchange points for a copy starting from over **two years** of the original release date.

	BiLSTM	GCGCN-BERT	MGRE-BERT	Ground Truth

Fig. 6. Case study of an example from the development set of DocRED. Different entity types are distinguished by color. The red arrow indicates the correct relation triple and the green arrow indicates that the relation triple is not in the standard answer. The relations shown in the figure are defined as follows. P123: *publisher*; P172: *ethnicgroup*; P178: *developer*; P179: *series*; P400: *platform*; P495: *countryoforigin*; P571: *inception*. (Color figure online)

between *developer* and *publisher* has similar semantics, which indicates that GCGCN-BERT is unable to handle details well with weak reasoning ability.

Only MGRE-BERT predicts all the correct triples, but it is also produces four answers which are not given by the ground truth. After careful analysis of the document, we find that except for (*Monster Games*, *developer*, *Excite Truck*), the other three answers can be obtained by reasoning, which means they are correct but missing from the DocRED dataset annotation. It indicates that MGRE-BERT has the ability to reason over multiple sentences. Specially, the triples that GCGCN predicts wrongly are predicted correctly by MGRE-BERT, which verifies the effectiveness of appropriate fusion of multi-granularity information and demonstrates the superiority of our proposed method.

5 Conclusion and Future Work

This paper proposes a multi-granularity relation extraction model MGRE, which has competitive performance for document-level RE. Experimental results demonstrate that MGRE can improve the representation of entity-level and sentence-level semantic information and further fuse with document-level semantic information to achieve a better improvement in RE.

References

1. Bordes, A., Usunier, N., García-Durán, A., et al.: Translating embeddings for modeling multi-relational data. In: NeurIPS (2013)
2. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: document-level neural relation extraction with edge-oriented graphs. In: EMNLP (2019)

3. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL, pp. 4171–4186 (2019)
4. Goyal, P., Dollár, P., Girshick, R., et al.: Accurate, large minibatch SGD: training imagenet in 1 hour (2018)
5. Gu, J., Sun, F., Qian, L., et al.: Chemical-induced disease relation extraction via convolutional neural network. *Databases* **2017**, bax024 (2017)
6. Gupta, P., Rajaram, S., Schütze, H., et al.: Neural relation extraction within and across sentence boundaries. In: AAAI, pp. 6513–6520 (2019)
7. Kim, K., Hur, Y., Kim, G., et al.: GREG: a global level relation extraction with knowledge graph embedding. *Appl. Sci.* **10**(3) (2020)
8. Li, B., Ye, W., Sheng, Z., et al.: Graph enhanced dual attention network for document-level relation extraction. In: ACL, pp. 1551–1560 (2020)
9. Lin, Y., Shen, S., Liu, Z., et al.: Neural relation extraction with selective attention over instances. In: ACL (2016)
10. Lin, Z., Feng, M., dos Santos, C.N., et al.: A structured self-attentive sentence embedding. In: ICLR (2017)
11. Liu, Y., Wei, F., Li, S., et al.: A dependency-based neural network for relation classification. In: ACL, pp. 285–290 (2015)
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
13. Lu, C., Zhang, R., Sun, K., Kim, J., Zhang, C., Mao, Y.: Anaphor assisted document-level relation extraction (2023)
14. Ma, Y., Wang, A., Okazaki, N.: DREEAM: guiding attention with evidence for improving document-level relation extraction. In: EACL (2023)
15. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1105–1116 (2016)
16. Nan, G., Guo, Z., Sekulic, I., et al.: Reasoning with latent structure refinement for document-level relation extraction. In: ACL, pp. 1546–1557 (2020)
17. Nguyen, D.Q., Verspoor, K.: Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. In: The 18th Workshop on Biomedical Natural Language Processing (2018)
18. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
19. dos Santos, C.N., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. In: ACL (2015)
20. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**(11), 2673–2681 (1997)
21. Socher, R., Huval, B., Manning, C.D., et al.: Semantic compositionality through recursive matrix-vector spaces. In: EMNLP (2012)
22. Tan, Q., He, R., Bing, L., Ng, H.T.: Document-level relation extraction with adaptive focal loss and knowledge distillation. In: ACL (2022)
23. Tang, H., et al.: HIN: hierarchical inference network for document-level relation extraction. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12084, pp. 197–209. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47426-3_16
24. Verga, P., Strubell, E., McCallum, A.: Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In: NAACL, pp. 872–884 (2018)
25. Wang, H., Focke, C., Sylvester, R., et al.: Fine-tune BERT for docRED with two-step process. *ArXiv* **abs/1909.11898** (2019)
26. Wang, L., Cao, Z., de Melo, G., et al.: Relation classification via multi-level attention CNNs. In: ACL (2016)

27. Xie, Y., Shen, J., Li, S., Mao, Y., Han, J.: Eider: empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In: Findings of the Association for Computational Linguistics: ACL 2022 (2022)
28. Xu, B., Wang, Q., Lyu, Y., et al.: Entity structure within and throughout: modeling mention dependencies for document-level relation extraction (2021)
29. Xu, W., Chen, K., Mou, L., Zhao, T.: Document-level relation extraction with sentences importance estimation and focusing. In: NAACL (2022)
30. Yao, Y., Ye, D., Li, P., et al.: DocRED: a large-scale document-level relation extraction dataset. In: ACL, pp. 764–777 (2019)
31. Zeng, D., Liu, K., Lai, S., et al.: Relation classification via convolutional deep neural network. In: ACL, pp. 2335–2344 (2014)
32. Zeng, S., Xu, R., Chang, B., et al.: Double graph based reasoning for document-level relation extraction. In: EMNLP (2020)
33. Zhang, L., Su, J., Chen, Y., Miao, Z., Zijun, M., Hu, Q., Shi, X.: Towards better document-level relation extraction via iterative inference. In: EMNLP (2022)
34. Zhou, H., Xu, Y., Yao, W., et al.: Global context-enhanced graph convolutional networks for document-level relation extraction. In: ACL, pp. 5259–5270 (2020)



Improving Zero-Shot Information Retrieval with Mutual Validation of Generative and Pseudo-Relevance Feedback

Xinran Xie^{1,2}(✉) , Rui Chen^{1,2} , TaiLai Peng^{1,2} , Dekun Lin^{1,2} , and Zhe Cui^{1,2}(✉)

¹ Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610213, China
cuizhe@casit.com.cn

² University of Chinese Academy of Sciences, Beijing 101408, China
xie Xinran19@mails.ucas.ac.cn

Abstract. Information retrieval systems often suffer from poor performance due to brief or ambiguous user queries. A direct and effective method is to expand these queries by incorporating additional information. Traditional Pseudo-Relevance Feedback (PRF) approaches enhance queries by extracting information from the top- k retrieved documents during the initial retrieval, with their effectiveness closely correlated to retrieval quality. Meanwhile, recent studies on Generative-Relevance Feedback (GRF) utilize Large Language Models (LLMs) to capture the latent search intent behind user queries, but may generate corpus-irrelevant contexts due to the inherent issues of hallucination in LLMs. To mitigate the limitations and maintain the advantages of both PRF and GRF, we propose a zero-shot sentence-level mutual validation framework. Specifically, we enrich the generative relevance feedback by synthesizing various prompting strategies. The mutual validation process incorporates a comprehensive scoring mechanism that considers both consistency and relevance dimensions, facilitating alignment between GRF and PRF while retaining query-relevant information. Lastly, our fine-grained dual-filtering approach ensures relevant and reliable sentences for robust query expansion. By conducting extensive analytical experiments on four low-resource datasets, we showcase the effectiveness of our proposed method compared to existing approaches, particularly in resource-constrained settings.

Keywords: Information retrieval · Query expansion · Large language models · Pseudo-relevance feedback

1 Introduction

In the realm of Information Retrieval (IR), the primary goal is to efficiently retrieve pertinent documents from an extensive corpus in response to user

queries. It is a core component of modern search engines and question-answering systems. Despite their importance, IR systems often struggle with short or vague user queries, which can lead to poor retrieval performance, especially in low-resource domains like medicine and biology. In such domains, the scarcity of annotated data poses a significant obstacle to training models that can precisely capture user intents. Query expansion has emerged as a widely used technique that improves retrieval performance by adding additional information to the user’s original query. The expanded query may be able to retrieve relevant documents that had no lexical overlap with the original query.

Traditional query expansion methods are typically based on Pseudo-Relevance Feedback (PRF) [1–4], which augment the query with information extracted from the top-k documents retrieved during an initial retrieval pass. The PRF-based approach assumes that the top-k retrieved documents are relevant to the query. However, in situations where the query is short or ambiguous, the initially retrieved documents may not align perfectly with the original query. With the advent of Large Language Models (LLMs), researchers have harnessed their powerful semantic understanding capabilities and vast knowledge repositories to generate valuable query expansions [5–7]. These extensions, known as Generative-Relevance Feedback (GRF), play a crucial role in capturing the underlying search intent behind user queries, enabling more efficient information retrieval. However, they can generate content that deviates from the existing corpus, a phenomenon termed “Generative Hallucination” [8].

Table 1 shows the results for the query: “Do spiders eat other animals?”. PRF, while successful in avoiding hallucinations by recalling documents from the corpus, may still retrieve sentences that are unrelated to the user’s intent, such as “animals that prey on spider”. In contrast, LLMs may generate sentences that are completely absent from the corpus, such as “various hunting methods of spider” and “benefits of spiders to human”. Therefore, more optimal approaches [9, 10] are to leverage the merits of both methods, considering the authenticity of PRF alongside the comprehensiveness of GRF. Direct integration, combining retrieved and generated documents from Table 1 without modifications, can potentially introduce noise into the IR system, as shown in the Q2D-Multi method in Table 3. The grey text in Table 1 highlights irrelevant sentences that may hinder retrieval performance.

Thus, we propose a novel strategy called Mutual Validation for Sentence-level Filtering (MVSF), aiming to balance the advantages of both PRF and GRF approaches while effectively eliminating irrelevant information unrelated to the query and corpus. To this end, we initially focus on synthesizing various prompt strategies to assist LLMs in generating diverse contextual information that covers the underlying search intent. Motivated by the intuition that a sentence is credible when supported by information from two sources, we build a framework for mutual verification at the sentence level. To be more specific, we first design a generated sentence filter that assesses the alignment between generated sentences and retrieved sentences using consistency scores. The calculation of consistency scores takes into account the relative importance of each sentence in the retrieved

Table 1. Examples from the MS MARCO dataset. Green texts are the overlapping words between ground truth and pseudo-documents. Gray texts represents sentences that differ significantly from the ground truth and can be successfully filtered out by our method.

Query	Do spiders eat other animals?
Retrieved documents	Most species trap small insects and other spiders in their webs and eat them. A few large species of spiders prey on small birds and lizards. But to many animals, spiders look delicious. Insectivores include bats and shrews, among hundreds of insect-and spider-eating species. Some types of monkeys also occasionally appreciate a spider snack.
Generated documents	Spiders are carnivorous creatures and their diet primarily consists of insects, although some species of spiders are known to consume larger prey such as lizards, frogs, and even small birds. ... Orb-weaving spiders construct intricate webs to ensnare flying insects, while jumping spiders rely on their agility and keen eyesight to pounce on unsuspecting prey. In agricultural settings, spiders help to reduce pest populations, thus benefiting crop production. In conclusion, spiders are carnivorous creatures that feed on a variety of other animals, primarily insects.
Ground truth	Most species trap small insects and other spiders in their webs and eat them. A few large species of spiders prey on small birds and lizards. One species is vegetarian, feeding on acacia trees. Some baby spiders eat plant nectar. In captivity, spiders have been known to eat egg yoke, bananas, marmalade, milk and sausages.

sentence set. Consequently, higher priority is given to retaining generated sentences that show greater consistency with key retrieved sentences. This design significantly enhances the effectiveness of filtering out generated sentences that deviate from the corpus. Furthermore, we have designed a retrieved sentence filter that leverages the generated sentences as references. The filter indirectly filters out retrieved sentences that are not relevant to the query intent based on consistency scores. Additionally, we enhance our mutual validation framework by incorporating relevance scores to directly assess the semantic match between sentences and queries. Finally, we filter out sentences with low scores by considering both the consistency scores and the relevance scores. In this way, our method ensures a high level of consistency between the extracted sentences and the corpus while also closely aligning with the specific requirements of the query.

To evaluate the proposed method, experiments are conducted on four low-resource datasets. The results demonstrate that our proposed method signifi-

cantly outperforms the baselines. In summary, the contributions of this paper are primarily as follows:

- 1) To facilitate a more exhaustive exploration of the knowledge space of the language model, we flexibly incorporate various prompting strategies to further diversify the expanded queries.
- 2) We propose a novel query expansion approach that preserves the consistency of traditional PRF methods with the corpus and the comprehensiveness of emerging GRF methods. The framework is able to ease the constraints of generating and retrieving context, enabling the provision of high-quality context for query expansion.
- 3) In low-resource scenarios, our method performs well even in zero-shot settings and can serve as a plug-and-play component adaptable to a variety of information retrieval systems.

2 Related Work

Query expansion is vital for enhancing information retrieval by bridging the language gap between queries and documents in the corpus. The roots of query expansion can be traced back to [1] in 1960, where it was used for indexing and searching within a mechanized library system.

2.1 Pseudo-Relevance Feedback

Early query extension work focused on Pseudo-Relevance Feedback (PRF), including RM3 [2], AxiomaticQE [3]. The idea behind relevance feedback is to incorporate the users feedback in the retrieval process so as to improve the final result [4]. Obtaining actual user feedback can be challenging, so PRF approaches used the results from an initial retrieval as a substitute, assuming these documents were relevant to extend the original query and boost retrieval performance. With the advancement of language models, there has been a shift towards using dense vector representations for PRF like ColBERT-PRF [11] and BERT-QE [12, 13], which typically require training or fine-tuning a model and still depend on the first retrieval’s performance. In contrast, our work takes advantage of the inherent capabilities of LLMs in zero-shot scenarios, making them more suitable for low-resource tasks.

2.2 Generative-Relevance Feedback

Large language models (LLMs) such as GPT-3 [14] and LLaMA [15] are trained on vast collections of text and have billions of parameters. LLMs enhance tasks effectively by utilizing just a description and a few examples without requiring updates to parameters [16]. Other studies [17, 18] have discovered that LLMs are powerful zero-shot reasoners. Consequently, LLMs have seen broad use in natural language processing, notably in refining information retrieval. Applications

include facet generation [19] and document expansion [20], as well as zero-shot document reranking [21]. Notable work has focused on using LLMs for query-specific reasoning to enhance retrieval performance. Taking inspiration from PRF, [5] developed the concept of “Generative-Relevance Feedback (GRF)”, a way to prompt LLMs to generate contexts related to the initial query, thereby extending the query itself. Q2D [6] introduces a “query document prompt” framework that utilizes the semantic understanding and generative potential of LLMs to enrich the original query. [7] investigates various prompting strategies, including Chain of Thoughts (CoT) [17] and PRF-enhanced document generation, to further extend queries.

2.3 Hybrid Method

Researchers in [9] have investigated the advantages of GRF and PRF, noting that GRF provides additional external context beyond the initial retrieval, while PRF aligns the query more closely with the target corpus. So they propose integrating both generative and pseudo-relevance feedback signals in the ranking process to leverage the strengths of each feedback type. On the contrary, [10] argue against blindly combining both methods without careful consideration. They propose a query expansion framework that employs LLMs for mutual validation.

Our approach shares a similar starting point with [10], leveraging the context generated by GRF and PRF to mutually verify each other. However, we distinguish ourselves by integrating diverse human-made prompting strategies, as shown in Table 2, to enhance the quality of GRF. Furthermore, we introduce consistency and relevance scores to effectively filter out noise at a more granular sentence level, resulting in enhanced performance across multiple datasets.

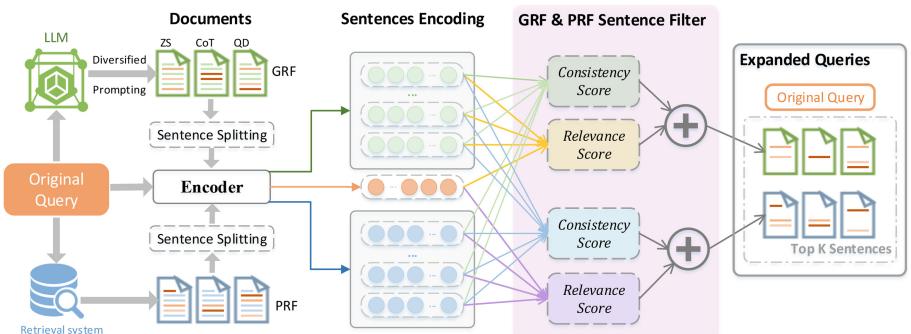


Fig. 1. Overall structure. The process of mutual validation between generated and pseudo-relevance feedback at the sentence level ensures that the retained sentences are highly correlated with the query and consistent with the corpus.

3 Overview

We aim to expand an initial user query Q into an expanded query Q' to improve the performance of IR systems. Our expansion strategy as shown in Fig. 1 employs two complementary techniques. GRF involves crafting prompts that encourage a LLM to produce responses that augment the original query, detailed in Sect. 3.1. In contrast, PRF improves the query by incorporating terms from the top-ranked documents. For this, we employ the classic retrieval algorithm BM25 [22], noted for its high efficiency and strong performance in zero-shot scenarios [23]. To enhance the synergy between GRF and PRF, we propose a mutual validation algorithm that operates at the sentence level. This algorithm is designed to filter out irrelevant sentences, thereby refining the expanded query Q' that is most likely to improve retrieval results, detailed in Sect. 3.2.

3.1 Diversifying GRF Through Flexible Prompting Strategies

Researchers have shown that the effectiveness of LLMs is greatly influenced by the design of the prompts [7, 17]. Various prompting strategies, such as zero-shot learning (ZS) [6], chain-of-thought (CoT) [7], and question decomposition (QD) [10], can be employed to generate contexts that are closely related to the original query. However, these studies typically examine the impact of a single prompt on documents generation. When multiple documents are generated from a single prompt, they tend to exhibit a uniform token distribution [24], which may not adequately address complex and diverse user queries.

Therefore, we supply diverse prompts to enhance the variability of the generated documents, as shown in Table 2. This straightforward approach effectively modifies the token distribution during generation.

Table 2. Diversified Prompting Strategies.

Method	Prompt
Zero-shot	Write passages answer the following query: {Q}
CoT	Answer the following query: {Q}. Give the rationale before answering.
Question Decomposition	What sub-queries should be searched to answer the following query: {Q}. I will generate the sub-queries and write passages to answer these generated queries:

3.2 Mutual Validation for Sentence-Level Filtering

Unlike the approach proposed by [10], our approach primarily emphasizes sentence-level filtering, the overall algorithm as shown in Algorithm 1.

Sentence Splitting. To effectively isolate and evaluate sentences, we utilize the Natural Language Toolkit (NLTK) to segment the documents into discrete sentences, denoted as $S_G = \{sg_1, sg_2, \dots, sg_n\}$ for GRF and $S_R = \{sr_1, sr_2, \dots, sr_m\}$ for PRF, where sg_i and sr_j represent an individual sentence. This segmentation facilitates the subsequent filtering process, which in turn enhances the relevance of the information provided to the user query.

Vector Representation. We utilize an off-the-shelf dense representation model, denoted as $\text{Encoder}(\cdot)$, to compute the representation of each sentence:

$$g_i = \text{Encoder}(sg_i), r_j = \text{Encoder}(sr_j), q = \text{Encoder}(Q), \quad (1)$$

where g_i , r_j , and q correspond to the vectors of the generated sentence, retrieved sentence, and query, respectively.

Generated Sentence Filter. We design a comprehensive scoring mechanism to evaluate each generated sentence. The total score $ts(g_i)$ is a combination of two factors: the relevance of the generated sentence to the original query and its consistency with all retrieved sentences. The set of vectors from all retrieved sentences is represented by $V_R = \{r_1, r_2, \dots, r_m\}$. We control the balance between these two factors using a parameter $\alpha \in [0, 1]$. The scoring formula is as follows:

$$ts(g_i) = \underbrace{\alpha \cdot cs(g_i, V_R)}_{\text{consistency score}} + \underbrace{(1 - \alpha) \cdot rs(g_i, q)}_{\text{relevance score}}. \quad (2)$$

Consistency Score. We measure the consistency score $cs(g_i, V_R)$ by calculating the cosine similarity (denoted as $\text{sim}(\cdot)$) between a generated sentence g_i and all retrieved sentences. In other words, assuming that the retrieved sentence r_j is consistent with the corpus, we calculate the probability $p(g_i|r_j)$ of the generated sentence g_i being consistent with the corpus. Considering the limitations of the retrieval system's performance, it is unrealistic to expect that every retrieved sentence will be relevant to the user's query. Therefore, assigning equal weight to each sentence would be inappropriate. Consequently, we factor in the relative importance of each retrieved sentence within the total set of retrieved sentences, denoted as $p(r_j)$. A sentence that is very similar to other sentences in the set is considered important because it captures the common information. So the final

consistency scoring formula is as follows:

$$cs(g_i, V_R) = \frac{1}{m} \sum_{j=1}^m p(g_i|r_j) \cdot p(r_j), \quad (3)$$

$$p(g_i|r_j) = sim(g_i, r_j), \quad (4)$$

$$p(r_j) = \frac{1}{m-1} \sum_{i=1, i \neq j}^m sim(r_j, r_i). \quad (5)$$

As can be seen from Eq. 3, we give priority to generated sentences that are highly correlated with high-importance retrieval sentences.

Relevance Score. We also calculate the cosine similarity of each generated sentence to the original query as:

$$rs(g_i, q) = sim(g_i, q). \quad (6)$$

Filtering Mechanism. To enhance the generalizability of our model across diverse datasets, we have not employed the conventional method of directly retrieving the top-k sentences. Instead, we utilize a standardization process that employs the z-score to select sentences with high scores from S_G , resulting in the formation of a new set of sentences denoted as G^* , as follows:

$$G^* = \left\{ sg_i \in S_G \mid \frac{ts(g_i) - \mu}{\sigma} > 0 \right\}, \quad (7)$$

where $ts(g_i)$ is the score of the generated sentence sg_i , μ is the mean score, and σ is the standard deviation of the scores. This method offers greater degrees of freedom and eliminates the need for adjusting the hyperparameter k. Moreover, it showcases robust generalization capabilities across a wide range of datasets.

Retrieved Sentence Filter. Similar to the method used in the previous section **Generated Sentences Filter**, we also score each retrieved sentence $ts(r_i)$ by calculating the relevance score with query and the consistency score with all generated sentences. The set of vectors from all generated sentences is represented by $V_G = \{g_1, g_2, \dots, g_n\}$. Therefore, the analysis process is not repeated here and is represented as follows:

$$ts(r_i) = \underbrace{\beta \cdot cs(r_i, V_G)}_{\text{consistency score}} + \underbrace{(1 - \beta) \cdot rs(r_i, q)}_{\text{relevance score}}. \quad (8)$$

Consistency Score. By calculating the similarity between each retrieved sentence and all generated sentences, denoted as $p(r_i|g_j)$, we indirectly evaluate the extent to which the retrieved sentence aligns with the true intention of the original query. It also considers the significance of each generated sentence, denoted

as $p(g_j)$, thereby mitigating the influence of noisy sentences. Therefore, the calculation process of $cs(r_i, V_G)$ is as follows:

$$cs(r_i, V_G) = \frac{1}{n} \sum_{j=1}^n p(r_i|g_j) \cdot p(g_j), \quad (9)$$

$$p(r_i|g_j) = \text{sim}(r_i, g_j), \quad (10)$$

$$p(g_j) = \frac{1}{n-1} \sum_{i=1, i \neq j}^n \text{sim}(g_j, g_i). \quad (11)$$

Relevance Score. The direct relevance between the retrieved sentence and the query is also determined using the cosine distance, as follows:

$$rs(r_i, q) = \text{sim}(r_i, q). \quad (12)$$

Filtering Mechanism. We select the retrieved sentences with high scores from S_R to create a new set, denoted as R^* . As shown in Eq. 13, μ and σ represent the mean and standard deviation of the scores across all sentences, respectively.

$$R^* = \left\{ sr_i \in S_R \mid \frac{ts(r_i) - \mu}{\sigma} > 0 \right\}. \quad (13)$$

3.3 Query Expansion for Retrieval

After mutual validation, we integrate the selected generative-relevance sentences G^* and pseudo-relevance sentences R^* with the original query Q to perform the final retrieval task. More formally, the objective for query expansion can be formalized as:

$$\underset{Q'}{\text{argmax}} \mathcal{M}(Q', RS), \text{ where } Q' = \text{Concat}(Q, Q, Q, Q, Q, R^*, G^*). \quad (14)$$

Here, \mathcal{M} denotes the evaluation metric (e.g., recall, nDCG) of the retrieval performance and RS denotes the retrieval model BM25. Since R^*, G^* may be verbose, we repeat the original query five times to upweigh their relative importance. This is the same as the trick employed by [6].

Algorithm 1. Mutual validation for sentence-level filtering

1: **Input:** Query Q , segmented generated sentences $S_G = \{sg_1, sg_2, \dots, sg_n\}$ and retrieval sentences $S_R = \{sr_1, sr_2, \dots, sr_m\}$
2: **Output:** Filtered generated sentences G^* and filtered retrieval sentences R^*
3: **Step 1: Sentence Encoding**
4: $V_G \leftarrow \{\text{Encoder}(sg_i) \mid sg_i \in S_G\}$
5: $V_R \leftarrow \{\text{Encoder}(sr_i) \mid sr_i \in S_R\}$
6: $q \leftarrow \text{Encoder}(Q)$
7: **Step 2: Generated Sentence Filtering**
8: Initialize $G^* \leftarrow \emptyset$
9: **for** each $g_i \in V_G$ **do**
10: **for** each $r_j \in V_R$ **do**
11: Consistency Score: $cs(g_i, V_R) \leftarrow p(g_i|r_j) \cdot p(r_j)$
12: **end for**
13: Relevance Score: $rs(g_i, q) \leftarrow \text{sim}(g_i, q)$
14: Total Score: $ts(g_i) \leftarrow \alpha \cdot cs(g_i, V_R) + (1 - \alpha) \cdot rs(g_i, q)$
15: **if** $\frac{ts(g_i) - \mu_G}{\sigma_G} > 0$ **then**
16: $G^* \leftarrow G^* \cup \{sg_j \in S_G\}$
17: **end if**
18: **end for**
19: **Step 3: Retrieved Sentence Filtering**
20: Initialize $R^* \leftarrow \emptyset$
21: **for** each $r_i \in V_R$ **do**
22: **for** each $g_j \in V_G$ **do**
23: Consistency Score: $cs(r_i, V_G) \leftarrow p(r_i|g_j) \cdot p(g_j)$
24: **end for**
25: Relevance Score: $rs(r_i, q) \leftarrow \text{sim}(r_i, q)$
26: Total Score: $ts(r_i) \leftarrow \beta \cdot cs(r_i, V_G) + (1 - \beta) \cdot rs(r_i, q)$
27: **end for**
28: **if** $\frac{ts(r_i) - \mu_R}{\sigma_R} > 0$ **then**
29: $R^* \leftarrow R^* \cup \{sr_i \in S_R\}$
30: **end if**
31: **Return** G^* and R^*

4 Experimental Setup

4.1 Datasets and Metrics

For zero-shot information retrieval evaluation, we use four low-resource datasets from the BEIR benchmark [23], a heterogeneous benchmark covering diverse sub-tasks such as question answering, tweet retrieval, biomedical IR, and entity retrieval:

- **TREC-COVID** [25]: TREC-COVID employs the established TREC framework to create test collections for evaluating IR systems, focusing on a document set that includes recent publications and preprints about COVID-19.
- **MS MARCO** [26]: MS MARCO is a large-scale machine reading comprehension resource constructed to advance deep learning development in search

applications. We choose the passage dataset as our experimental scenario and take the first 100 queries from the dev group as the test queries.

- **Touche-2020** [27]: It aims to assist users in finding compelling arguments from online debate forums for use in discussions on contentious issues.
- **NFCorpus** [28]: NFCorpus is a full-text English retrieval data set for Medical Information Retrieval. The queries are taken from health topics described in laymans English on the non-commercial website.

Following previous work [6, 7], we use nDCG@10 to measure top-heavy ranking metrics, while Recall@1K focuses on the IR system’s ability to successfully retrieve relevant documents within the first 1,000 results.

4.2 Hyperparameters

We implement all the experiments with PyTerrier [29], a Python library helps conduct information retrieval experiments. For the BM25 retriever, we use the default parameters ($b = 0.75$, $k_1 = 1.2$, $k_3 = 8.0$) provided by PyTerrier. For all the LLM-based baselines, we use the gpt-3.5-turbo API [14] provided by OpenAI to generate contextual documents. The generation parameters are set as $\text{temperature} = 0.7$. We use the Sentence-Transformers/all-MiniLM-L6-v2 [30] as the text encoder, which maps sentences or paragraphs to a 384 dimensional dense vector space. we select five candidate documents for PRF and use three prompts to generate three candidate documents for GRF. In Eq. 2 and 8, the initial values for parameters α and β are both set to 0.5. The influence of the values of the two parameters on the results will be discussed later.

4.3 Baselines

Since different works may have various experimental settings, the results of following baselines are reproduced under the unified experimental setting (Sect. 4.2), which slightly deviate from the results shown in original papers.

- **Traditional Query Expansion Methods.** This method directly treats the retrieved top-ranked document as an extension of the original query.
- **LLM-based Expansion Methods.** Follow [6, 7, 10], we investigated the effectiveness of various prompt methods for GRF, including Q2D-ZS, Q2D-CoT, and Q2D-QD, which are detailed in Table 2.
- **Hybrid Methods. Q2D-Multi** directly mixes PRF and GRF without any filtering operation. GRF is a mixture of documents generated by multiple prompt strategies. **MILL** [10] employs a single-prompt strategy for document generation and performs mutual validation with retrieved documents at the document level.

5 Results and Analysis

5.1 Main Results

Here we present our experimental results in Table 3. Our approach achieves the best or second-best performance on all metrics and datasets and is a more suitable zero-shot approach for a variety of practical applications.

Table 3. Overall experimental results on 4 datasets. The optimal results are highlighted in bold, while the suboptimal results are underscored. **+RSF** refers to the use of only the retrieved sentence filter. **+GSF** denotes the sole use of the generated sentence filter. **+MVSF** represents a hybrid approach that combines the retrieved and generated sentence filters.

Method	TREC-COVID		MS MARCO		Touche-2020		NFCorpus		Average
	nDCG@10	R@1k	nDCG@10	R@1k	nDCG@10	R@1k	nDCG@10	R@1k	
No expansion									
BM25	62.59	40.52	28.68	86.50	34.27	85.04	32.22	36.06	50.36
Traditional expansion									
PRF	62.19	42.37	21.79	79.50	33.94	82.18	31.90	59.46	51.67
PRF+ RSF	66.48^{+4.3}	44.92^{+2.5}	26.50^{+4.7}	84.50^{+5.0}	36.60^{+2.7}	85.49^{+3.3}	34.82^{+2.9}	63.08^{+3.6}	55.30
LLM-based expansion									
Q2D-ZS[6]	66.54	44.81	26.27	87.50	41.78	84.77	35.47	61.22	56.05
Q2D-ZS + GSF	70.80^{+4.3}	46.21^{+1.4}	28.68^{+2.4}	91.50^{+4.0}	41.21^{+0.6}	85.33^{+0.6}	35.10^{-0.4}	59.13^{-2.0}	57.25
Q2D-CoT	66.69	45.24	26.05	86.50	41.94	83.10	34.84	59.85	55.53
Q2D-CoT + GSF	69.49^{+2.8}	46.76^{+1.5}	27.11^{+1.1}	88.50^{+2.0}	41.02^{-0.9}	84.43^{+1.3}	35.34^{+0.5}	59.63^{-0.2}	56.54
Q2D-QD	68.04	44.90	23.35	89.50	38.87	84.03	34.70	61.23	55.58
Q2D-QD + GSF	72.55^{+4.5}	45.97^{+1.1}	26.47^{+3.1}	90.50^{+1.0}	40.17^{+1.3}	85.10^{+1.1}	34.90^{+0.2}	59.70^{-1.5}	56.92
Hybrid expansion									
MILL	72.48	47.67	26.09	84.50	42.02	84.94	36.67	65.81	57.52
Q2D-Multi	69.40	48.19	24.22	89.50	41.92	84.60	35.76	65.96	57.44
Q2D-Multi + MVSF	73.73^{+4.3}	48.76^{+0.6}	27.40^{+3.2}	90.50^{+1.0}	42.92^{+1.0}	85.49^{+0.9}	36.24^{+0.5}	65.74 ^{-0.2}	58.85

While traditional PRF-based query extension is effective, GRF-based methods demonstrate significantly improved retrieval performance, with the exception of nDCG@10 on the MS MARCO dataset. The limitation likely arises from the MS MARCO dataset’s structure, which includes just one relevant document for each query, affecting the top document’s ranking when additional generated contexts are introduced. However, GRF significantly enhances its R@1k metric by enriching the original query.

Integrating GRF with PRF generally enhances the R@1k metric, benefiting from the synergy between the two information sources. However, an unfiltered blend of the two can lead to a notable decrease in nDCG@10, as hallucinations inherent in GRF and noise from PRF adversely affect the ranking of the top-10 most relevant documents in retrieval tasks.

Our method employs sentence-level mutual validation to effectively filter out irrelevant sentences, thereby improving the relevance ranking of retrieved documents. Impressively, rather than diminishing recall performance, our filtering mechanism actually improves it in certain tasks. As shown in Fig. 2, even though our method excluded almost half of the sentences, the retrieval performance did not suffer. On the contrary, it improved, which underscores the effectiveness of our filtering strategy.

5.2 Ablation Study

MVSF Mechanism is Crucial. As shown in Table 3, applying our filtering mechanism solely to the traditional expansion (+RSF) results in a performance increase of approximately 2% to 5%. When this mechanism is applied to LLM-based expansion (+GSF), it generally performs very well on most datasets. However, we observed a decrease in performance on the NFCorpus dataset. The discrepancy may stem from the significant difference in distribution between PRF and GRF. The NFCorpus retrieval corpus is sourced from specialised medical texts rich with empirical data, whereas its LLM typically responds to informal inquiries without citing supporting literature. This observation reminds us that we should design prompt engineering for LLMs according to the data features of the corpus. For instance, implementing a prompt constraint like “Please cite medical literature to answer the question according to the query:”. We plan to explore this approach in forthcoming studies.

Scoring Mechanism is Reasonable. As shown in the Eq. 2 and Eq. 8, our scoring system encompasses both a consistency score and a relevance score. To assess the validity of this mechanism, we conducted experiments where we randomly eliminated half of the sentences, adhering to the distribution depicted in Fig. 2. The outcomes, presented in Table 4, indicate that random sentence filtering adversely affects retrieval efficacy, performing even worse than when sentences are not filtered.

Furthermore, we design experiments to evaluate how varying the parameters α and β within the scoring mechanism influences the overall retrieval performance. The value of parameter β signifies the balance between relevance

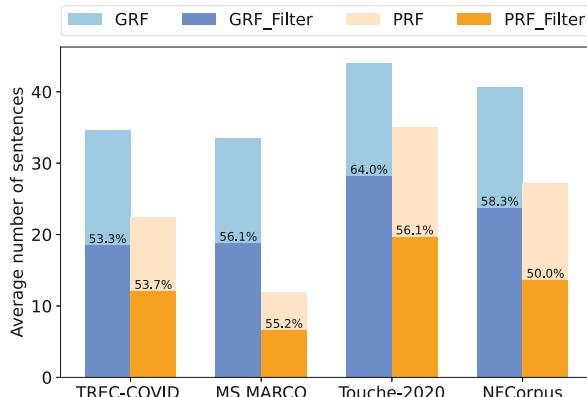


Fig. 2. We measured the average number of sentences generated and retrieved both before and after the application of our MVSF. The percentages displayed on the bar chart, like 53.3%, indicate the proportion of sentences remaining after the filter relative to the original count. The chart clearly shows that the filter has removed nearly half of the sentences.

Table 4. Randomly filtering sentences has a detrimental effect, even worse than if they were not filtered at all.

Method	TREC-COVID		MS MARCO		Touche-2020		NFCorpus	
	nDCG@10	R@1k	nDCG@10	R@1k	nDCG@10	R@1k	nDCG@10	R@1k
w/o MVSF	69.40	48.19	24.22	89.50	41.92	84.60	35.76	65.96
Random filtering	67.95	47.76	21.56	89.50	41.51	83.90	34.77	64.67
Our method	73.73	48.76	27.40	90.50	42.92	85.49	36.24	65.74

scores and consistency scores within the retrieved sentence filter. We fixed parameter α within the generated sentence filter and varied β during our experiments. To ensure the experiments' reliability, we selected three different settings for α : 0.2, 0.5, and 0.8. The results, presented in Fig. 3(a), demonstrate a stable trend: both the relevance and consistency scores are essential, and an imbalance in their proportions adversely affects performance. Using a similar experimental approach, we analyze the role of parameter α in the generated sentence filter. As depicted in Fig. 3(b), a lower value of α correlates with enhanced performance, highlighting the importance of the relevance score in the sentence filtering process.

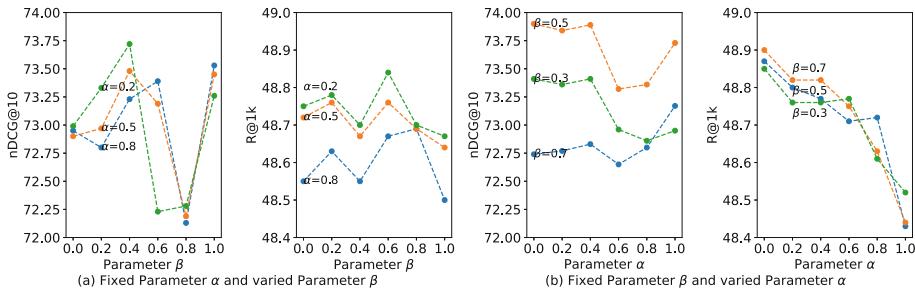


Fig. 3. Change the ratio of parameters α and β on TREC-COVID dataset.

Varying the Number of Candidate Documents. In the above experiments, we used the method described in Sect. 3.1 to generate 3 documents, and the number of retrieved documents was initially set to 5. In this section, we employ the control variable method to adjust the number of candidate documents. Specifically, we vary the number of candidate retrieval documents while maintaining a constant number of candidate generated documents, and then we reverse the procedure.

From Fig. 4, we have observations: (1) As the number of document retrieval candidates increases, there is a corresponding rise in R@1k. However, nDCG@10 plateaus after reaching five candidate documents, suggesting that adding more

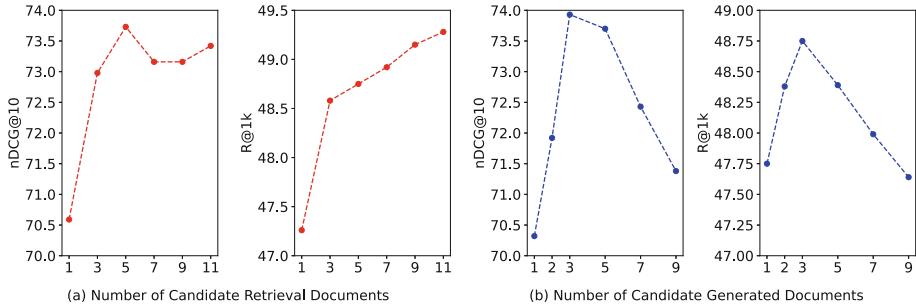


Fig. 4. Varying the number of candidate documents on TREC-COVID dataset.

candidates beyond this point introduces noise that hinders query expansion effectiveness. (2) Our approach utilizes three unique prompt strategies to generate candidate documents. Having more than three candidates implies strategy repetition, resulting in a significant decrease in nDCG@10 and R@1k scores. Employing a single prompt strategy may precipitate a substantial homogenization of content, which decreases the variety and richness of the information in GRF. Therefore, it becomes essential to integrate multiple prompt strategies to mitigate this issue.

6 Conclusion

In this paper, we present a novel sentence-level mutual validation framework that combines generative and pseudo-relevant feedback for query extension. Our approach involves synthesizing different prompt strategies to aid LLMs in generating diverse contextual information that covers the underlying search intent. We then introduce a sentence-level validation framework that leverages both retrieved and generated sentences to complement each other, addressing their respective limitations and enhancing the credibility of associated sentences. The experimental results showcase the robust performance of our approach, particularly in zero-shot settings. In addition, our approach can be used as a plug-and-play component that integrates seamlessly with a wide range of information retrieval systems, making it highly adaptable and versatile.

References

1. Maron, M.E., Kuhns, J.L.: On relevance, probabilistic indexing and information retrieval. *J. ACM (JACM)* **7**(3), 216–244 (1960)
2. Abdul-Jaleel, N., Allan, J., Bruce Croft, W., et al.: UMass at TREC 2004: novelty and hard. (189) (2004)
3. Fang, H., Zhai, C.: Semantic term matching in axiomatic approaches to information retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 115–122 (2006)

4. Rocchio, J.J.: Relevance feedback in information retrieval (1971). The full reference details should be specified here
5. Mackie, I., Chatterjee, S., Dalton, J.: Generative relevance feedback with large language models. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, pp. 2026–2031. Association for Computing Machinery (2023)
6. Wang, L., Yang, N., Wei, F.: Query2doc: query expansion with large language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, December 2023, pp. 9414–9423 (2023)
7. Jagerman, R., Zhuang, H., Qin, Z., et al.: Query expansion by prompting large language models (2023)
8. Huang, L., Yu, W., Ma, W., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions (2023)
9. Mackie, I., Chatterjee, S., Dalton, J.: Generative and pseudo-relevant feedback for sparse, dense and learned sparse retrieval (2023)
10. Jia, P., Liu, Y., Zhao, X., et al.: MILL: mutual verification with large language models for zero-shot query expansion (2023)
11. Wang, X., MacDonald, C., Tonello, N., et al.: ColBERT-PRF: semantic pseudo-relevance feedback for dense passage and document retrieval. ACM Trans. Web **17**(1), 1–39 (2023)
12. Zheng, Z., Hui, K., He, B., et al.: BERT-QE: contextualized query expansion for document re-ranking. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4718–4728, Online, November 2020. Association for Computational Linguistics (2020)
13. Zheng, Z., Hui, K., He, B., et al.: Contextualized query expansion via unsupervised chunk selection for text retrieval. Inf. Process. Manage. **58**(5), 102672 (2021)
14. Brown, T.B., Mann, B., et al.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS 2020, Red Hook, NY, USA. Curran Associates Inc (2020)
15. Touvron, H., Lavril, T., Izacard, G., et al.: LLaMA: open and efficient foundation language models (2023)
16. Zhao, W.X., Zhou, K., Li, J., et al.: A survey of large language models. *ArXiv*, abs/2303.18223 (2023)
17. Kojima, T., Gu, S.S., Reid, M., et al.: Large language models are zero-shot reasoners (2023)
18. Besta, M., Blach, N., Kubicek, A., et al.: Graph of thoughts: solving elaborate problems with large language models (2024)
19. Samarinis, C., Dharawat, A., Zamani, H.: Revisiting open domain query facet extraction and generation. In: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2022, pp. 43–50. Association for Computing Machinery, New York (2022)
20. Nogueira, R., Yang, W., Lin, J., et al.: Document expansion by query prediction (2019)
21. Ma, X., Zhang, X., Pradeep, R., Lin, J.: Zero-shot listwise document reranking with a large language model (2023)
22. Robertson, S., Walker, S., Jones, S., et al.: Okapi at TREC-3, pages 0–, January 1994
23. Thakur, N., Reimers, N., Rücklé, A., et al.: BEIR: a heterogenous benchmark for zero-shot evaluation of information retrieval models (2021)
24. Yu, W., Iter, D., Wang, S., et al.: Generate rather than retrieve: large language models are strong context generators (2023)

25. Voorhees, E.M., Alam, T., Bedrick, S., et al.: TREC-COVID: constructing a pandemic information retrieval test collection. *CoRR*, abs/2005.04474 (2020)
26. Nguyen, T., Rosenberg, M., Song, X., et al.: MS MARCO: a human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268 (2016)
27. Bondarenko, A., et al.: Overview of Touché 2021: argument retrieval. In: Candan, K.S., et al. (eds.) CLEF 2021. LNCS, vol. 12880, pp. 450–467. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85251-1_28
28. Boteva, V., Gholipour, D., Sokolov, A., et al.: A full-text learning to rank dataset for medical information retrieval (2016)
29. Macdonald, C., Tonellootto, N.: Declarative experimentation in information retrieval using PyTerrier. In: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, September 2020
30. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, November 2019



Entity Semantic Feature Fusion Network for Remote Sensing Image-Text Retrieval

Jianan Shui, Shuaipeng Ding, Mingyong Li^(✉), and Yan Ma^(✉)

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

Abstract. Recently, there has been remarkable progress in remote sensing image-text retrieval (RSITR), but in the past RSITR methods, researchers often try to extract features in images and texts from global and local perspectives, and the unique entity semantic contained in remote sensing images and texts rarely paid attention to, or even ignored. In this paper, we propose an Entity Semantic feature Fusion Network (ESFN), which uses the entity semantic in remote sensing images and texts to enhance the alignment degree and improve the retrieval accuracy. In the visual part, we propose a Scene Entity Filtering module (SEF), which can effectively extract significant entity semantic features from low-level feature maps. The Multi-level Adaptive Fusion module (MAF) adaptively selects the information of image features at different levels for feature fusion. In the textual part, we embed the entity semantic in the text into our textual feature extractor, so that it can have a good entity perception of remote sensing text. We designed a Text Phrase Enhancement module (TPE) to further extract and enhance entity semantic and alignment visual information in text. In addition, ESFN's experimental results on RSICD and RSITMD datasets show that R@1 and meanRecall (mR) reach 8.14, 22.16, 18.81 and 37.70 respectively, which verifies the model's perception of entity semantic in remote sensing images and texts. Through performance comparison, ablation study and visualization analysis, the effectiveness and superiority of this method are verified.

Keywords: Remote Sensing · Image-Text Retrieval · Entity Semantic

1 Introduction

In the era of information explosion, the rapid development of remote sensing technology provides rich data sources [14, 23] for the processing and analysis of geographic information. However, with the continuous accumulation of remote sensing image and text data, how to efficiently retrieve and manage these massive data has become an important challenge. As remote sensing image has the characteristics of multi-scale, multi-resolution and multi-target, the traditional cross-modal retrieval method is mainly aimed at natural scene image [5, 26], if it

is directly applied to remote sensing image, it is prone to problems such as single scale, incomplete feature extraction and wrong target matching, resulting in reduced retrieval effect. Therefore, researchers began to study the cross-modal retrieval methods of remote sensing data. Earlier methods extracted features from images and mapped them into the same space. Since remote sensing images have a wide shooting range and rich information, in addition to meaningful targets in the images, they also contain a large amount of redundant information and background information unrelated to description text. In order to better capture truly useful information and reduce interference, Yuan et al. [23] proposed a multi-scale visual self-attention module. The module integrates different scale visual features, which can effectively solve the multi-scale scarcity and target redundancy problems in remote sensing multi-modal retrieval tasks. Yuan et al. [25] uses the fusion of global and local features in remote sensing images to enhance visual representation. This method can accurately describe local features in remote sensing images, but additional knowledge is required. In order to solve the semantic confusion caused by remote sensing scene differences, Pan et al. [18] proposed a scene perception aggregation network to improve the retrieval effect by reducing the semantic confusion area in the semantic space. Ji et al. [9] proposed a knowledge-aided momentum contrastive learning retrieval method for nuance differences in remote sensing texts. These works have made great progress for remote sensing cross-modal retrieval.

Although these image-text retrieval methods have made great progress in targeting multi-scale, multi-resolution and multi-target features of remote sensing images, we find that the entity semantic samples in remote sensing are not well utilized. Previous works has attempted to obtain better image features by reducing redundant features [23] or from both global and local features [25], or to enhance alignment between images and texts from differences between remote sensing text descriptions [9], however, the entity features in remote sensing are more prominent than other features, and different remote sensing images and texts have different entity features. If we can accurately learn the characteristics of entity semantic in different remote sensing images and texts, then we can distinguish the different differences in remote sensing images and texts. Therefore, how to make good use of these unique entity semantic has become the focus of our research.

In order to make better use of entity semantic in remote sensing images and texts, we propose an efficient entity semantic feature fusion network (ESFN) for remote sensing image-text retrieval. The unique entity semantics in remote sensing images and text are extracted by using convolution kernels of different sizes, and an adaptive method is used to fuse each feature map in remote sensing image feature extraction.

The main contributions of our work are as follows:

- We propose a new entity semantic feature fusion network ESN, which extracts and utilizes entity semantic feature in image-text pairs of remote sensing to enhance the relationship between images and texts.

- The SEF module is used to extract the entity semantic in the salient feature, and the MAF module is used to adaptive fusion the features of different levels. The TPE module is designed to further extract the significant entity words and phrases in remote sensing texts so as to enhance text representation and align visual information.
- In addition, we propose to embed entity words in the textual feature extractor to effectively enhance its learning of key concepts. The effectiveness and superiority of ESFN are verified by experiments on RSICD and RSITMD.

2 Related Work

Although image-text retrieval based on natural images has made great progress, it is difficult to apply traditional methods to remote sensing images directly because of the characteristics of multi-scale, multi-granularity targets and complex scene types. To this end, Abdullah et al. [1] first proposed remote sensing image-text retrieval, which directly extracted features from remote sensing images and texts and mapped them into semantic space to achieve modal alignment. Cheng et al. [2] attempt to enhance the semantic consistency of the model through cross-modal interactions between images and text. Lv et al. [15] propose a cross-modal fusion correlation learning model to obtain complementary information between patterns and aggregate features. However, due to the large amount of non-semantic relevant information in remote sensing images, the truly useful image information only accounts for a part of the image. Yuan et al. [24] proposed an implicit supervision optimization method based on knowledge distillation to better capture these truly useful information and reduce interference information. Yuan et al. [23] proposed a visual self-attention module aiming at the characteristics of multi-scale and multi-granularity objects in remote sensing images, which fused features of different scales to strengthen visual representation and reduce redundancy. After that, researchers began to focus on the intrinsic connections in remote sensing image and text, and Yuan et al. [25] used the fusion of global and local features to enhance visual representations. The multi-level information dynamic fusion module is designed to realize the fusion of global and local visual features. This method can accurately describe local features in remote sensing images, but additional knowledge is required. Mi et al. [17] use knowledge graphs to enhance their textual representation and reduce the information gap between vision and text. Ji et al. [9] proposed a knowledge-assisted retrieval method for nuance-variance momentum contrast learning in remote sensing texts. For complex and similar scenes of remote sensing images, Pan et al. [18] combines visual features of different scales and granularity to obtain a visual aggregation representation with scene perception. Zheng et al. [27] proposed the scale-semanticjoint decoupling network to jointly solve scale decoupling and semantic decoupling in a unified model. Zhu et al. [28] combined training of segmentation and retrieval tasks can effectively alleviate the feature confusion problem caused by high noise in remote sensing scenes.

3 Methodology

In this section, we introduce our proposed ESFN method, as shown in Fig. 1. Subsequently, we describe visual feature extractor and textual feature extractor in detail in Sect. 3.1 and 3.2, the loss function we used is described in Sect. 3.3.

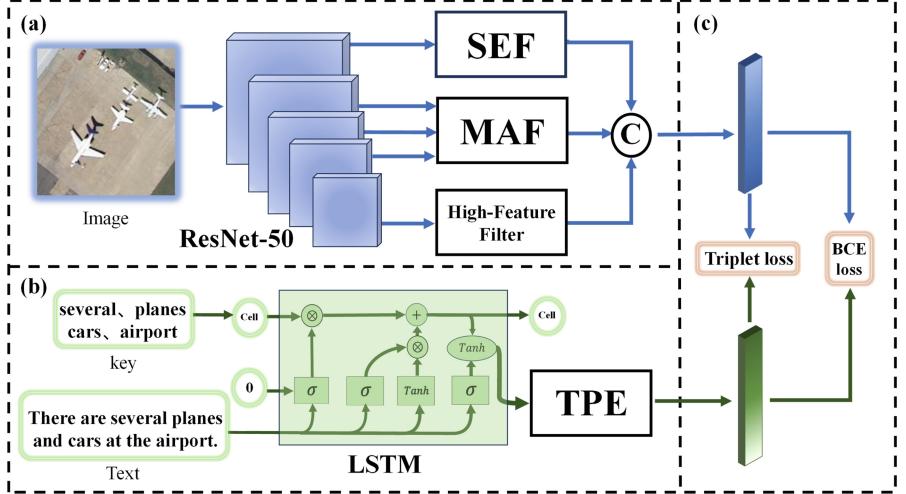


Fig. 1. Overview of our ESFN model, which consists of three parts: (a) visual extraction, including our proposed SEF module, which can effectively extract important entity semantic features from low-level feature mappings; The MAF module can adaptively fuse image features at different mid-levels. (b) textual extraction, including TPE module can further extract entity word and entity phrase semantic feature from remote sensing text. (c) similarity measurement.

3.1 Visual Feature Extractor

We use ResNet-50 [6] pre-trained on AID dataset [21] as the backbone for visual feature extraction in remote sensing image analysis. The ResNet is fine-tuned on AID dataset, and compared with the model fine-tuned on ImageNet [3], it shows better feature extraction capabilities tailored to remote sensing images. In addition, it is also good at capturing key features such as multi-scale and multi-granularity contained in remote sensing images. It is very effective to capture the features of remote sensing images.

When an image $I \in R^{H \times W \times C}$ undergoes convolutional network processing, the initial layer often produces image feature that is noisy but contains more textures and edges. To extract these fine-grained features. We used the previous salient feature of the ResNet, represented as I_{f0} , which contains more entity level semantic feature and can have a good capture of the entity. Similarly, we express the characteristics of ResNet's four layers layer1, layer2, layer3 and layer4 as $I_{fi} \in R^{H \times W \times C}$ ($i \in 1, 2, 3, 4$) with i ranging from 1 to 4, then individually input into the various specialized modules proposed in our method (Fig. 2).

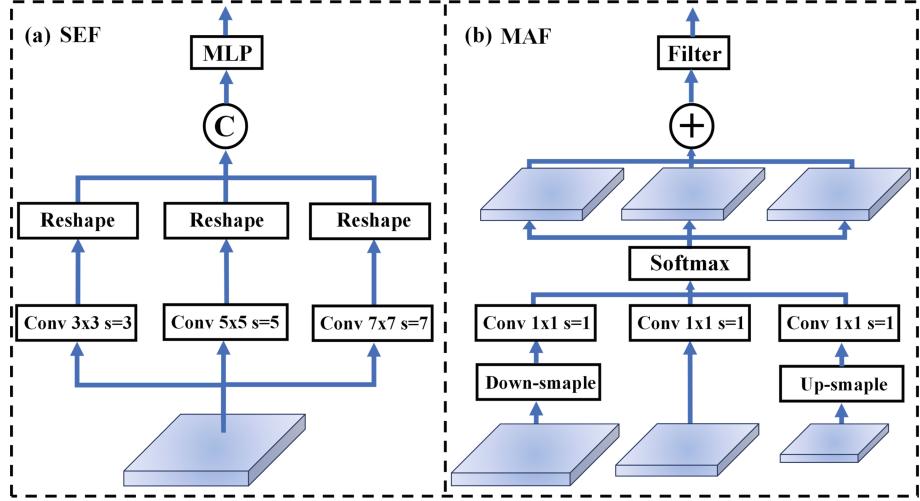


Fig. 2. Framework diagram of (a) Scene Entity Filtering module (SEF) and (b) Multi-level Adaptive Fusion module (MAF).

Scene Entity Filtering Module. The feature map of the image obtained at the shallow layer of the convolutional neural network usually contains abundant edge and details features, and with noise features. Similarly, in the remote sensing images, edge and detailed feature are crucial for extracting entity features. In order to make better use of these entity features, we propose a scene entity filtering module (SEF) aimed at make full use of the entity semantic feature in the shallow features of the convolutional network while reducing irrelevant noise. Since the entity features in remote sensing images have different sizes and different granularity size differences, we choose 3×3 , 5×5 , 7×7 convolution blocks to further extract the entity semantic of shallow features.

$$I_{low}^j = \text{Reshape} \left(\sigma \left(BN \left(\text{Conv}_{2d}^j (I_{f0}) \right) \right) \right), j = 3, 5, 7 \quad (1)$$

I_{low}^j represents the lower-level features obtained after three layers of convolution filtering of different sizes. $\text{Conv}_{2d}^j(\cdot)$ represents 2D convolution with different convolution nuclei and steps. The $\text{Reshape}(\cdot)$ represents reshaping the two dimensions (H, W) into the same dimension. σ and BN respectively represent the $ReLU$ function and batch normalization. We directly combine salient features at different scales as

$$V_{low} = \text{Avg}_{(1)}(\text{MLP}(\text{Concat}(I_{low}^3, I_{low}^5, I_{low}^7))) \quad (2)$$

where $I_{low}^j, j = 3, 5, 7$ represents multi-dimensional entity semantic features. After the extracted entity semantic features of three different sizes are spliced through a fully connected layer, the first dimension is finally averaged. $\text{Avg}_{(1)}$

represents the average of the first dimension. Thus we can get the $V_{low} \in R^{512}$ represents the low-level entity semantic features.

Multi-level Adaptive Fusion Module. The middle layer features in the convolutional network model have more semantic features than the shallow layer features, and the interference of noise feature is reduced to a certain extent. As for remote sensing images with various scales and granularity feature, the previous work on how to integrate semantic features at different levels usually involves channel splicing and filtering to obtain the merged features, or self-attention fusion and cross-attention fusion through attention mechanism [19]. These methods have certain limitations. Mainly because they fixed the proportion of different levels of feature fusion. Inspired by ASFF [13], we propose a multi-layer adaptive fusion module (MAF) to solve the shortcomings of the above methods. For the first three layers of features in ResNet, in order to better use the semantic level feature information contained in them. The first three features of ResNet, layer1, layer2 and layer3, are processed differently to maintain a consistent feature graph size. First, we use 1×1 convolution to deal with features of different depths in layer1, layer2, and layer3 layers, which can be expressed as

$$I_{mid}^1 = \text{Downsample}(\text{Conv}_{2d}^1(\text{layer}_1)) \quad (3)$$

$$I_{mid}^2 = \text{Conv}_{2d}^1(\text{layer}_2) \quad (4)$$

$$I_{mid}^3 = \text{Upsample}(\text{Conv}_{2d}^1(\text{layer}_3)) \quad (5)$$

$\text{Upsample}(\cdot)$ and $\text{Downsample}(\cdot)$ indicate up and down sampling. layer1 is processed by 1×1 convolution to activate the critical feature, then it is down-sampled, and then it is further cleaned by 1×1 convolution. Further cleaning of layer2 by 1×1 convolution; layer3 also use 1×1 convolution to process, then upsamples it, and then further cleans it after 1×1 convolution. In this way, we get three semantic features of the same size and dimension, but at different levels. And then we do channel compression as

$$I_{mid}^c = \text{Concat}(\text{Conv}_{2d}^c(I_{mid}^i)), i = 1, 2, 3 \quad (6)$$

We will concatenate the compressed feature maps, and then use a weight convolution layer to get the corresponding weight of each layer as

$$V_{weight} = \text{Softmax}(\text{Conv}_{2d}^w(I_{mid}^c)) \quad (7)$$

The $\text{Softmax}(\cdot)$ represents softmax function, which mutually exclusive selects the features of each layer, makes the sum of the weight ratio of the features of the three layers finally 1, this part can be expressed as follows:

$$V_{mid} = \text{Avg}_{(H,W)}(I_{mid}^1 \times V_{weight}^1 + I_{mid}^2 \times V_{weight}^2 + I_{mid}^3 \times V_{weight}^3) \quad (8)$$

Finally, the three layers of features are multiplied by their corresponding weights respectively, and then added, $\text{Avg}_{(H,W)}$ represents the average value of the (H,W) dimension, where $V_{mid} \in R^{512}$ represents the final adaptive mid features.

High Feature Filter Module. High-level feature maps in convolutional network models are often viewed as a global representation of the image with stronger semantic-level features, typically using a full connection layer to map high-level visual features to feature vectors of target size. Here, we filter it to the final visual feature dimension using 1×1 convolution, average its features from (H,W) dimension, expressed as

$$V_{high} = \text{Avg}_{(H,W)}(\text{Conv}_{2d}^1(\text{layer}_4)) \quad (9)$$

where $V_{high} \in R^{512}$ represents the high visual features.

Visual Aggregation Representation. In the end, we spliced and fused the features of three different levels, low level, mid level and high level, and use MLP to enhance non-linear mapping of visual features as

$$V_f = \text{MLP}(\text{Concat}(V_{low}, V_{mid}, V_{high})) \quad (10)$$

where $V_f \in R^{512}$ represents our final visual feature representation.

3.2 Textual Feature Extractor

For remote sensing text feature extraction, we choose to use long short-term memory (LSTM) [7] as the backbone of textual feature extraction. In remote sensing text, entity semantic is very important, different entity semantic can make us quickly perceive different scenes. In order to better grasp the prominent entity semantics in remote sensing text, we extract the important entity semantic words and phrases contained in the text as keywords from remote sensing text, and embed them in the memory unit initialized by LSTM, so that the LSTM network can better learn the keywords contained in the text when extracting text features, and avoid being forgotten. Then we use LSTM to extract remote sensing text features, and the formula is as follows:

$$T_{lstm} = \text{LSTM}(T_s, (H_0, T_{key})) \quad (11)$$

where T_s represents the original remote sensing text statement, H_0 represents the initialization of the hidden unit in LSTM, which is initialized to 0, and T_{key} represents the entity words and phrases with significant features that we extract from the original remote sensing text statement, and we embed them into the memory unit in LSTM.

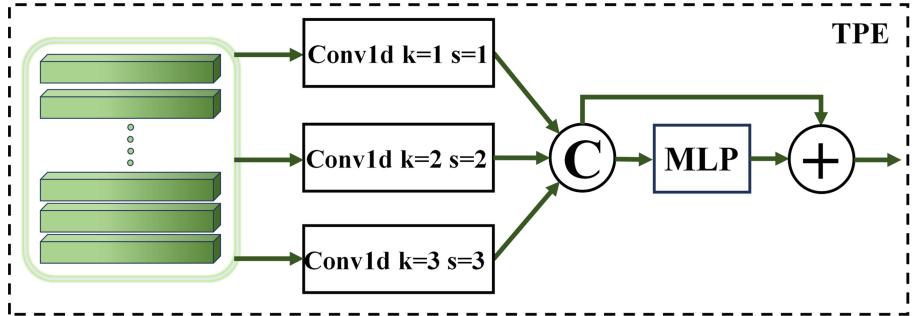


Fig. 3. Framework diagram of Text Phrase Enhancement Module (TPE).

Text Phrase Enhancement Module. In order to further enhance the perception of entity semantic words and words phrases in remote sensing texts, we propose the text phrase enhancement module (TPE) (Fig. 3) to learn entity semantic features contained in text. The TPE consists of three Conv1d with different sizes. By moving with different length steps, the text feature can be extracted at different scales, and the words and phrases in it can be further captured. The formula is as follows:

$$T_i = \sigma(BN(Conv_{1d}^j(T_{lstm}))), i = (1, 2, 3), j = (1, 2, 3) \quad (12)$$

where T_i represents text features obtained from three 1-dimensional convolution blocks, and $Conv_{1d}^j(\cdot)$ represents 1D convolution with different convolution nuclei and steps. Finally, text entity semantic of different sizes is connected through channel-level connection as

$$T_4 = Concat(T_1, T_2, T_3) \quad (13)$$

where T_4 indicates the fusion of text entity semantic features of different size. Finally, we use MLP to enhance the non-linear mapping of text features, and to ensure that no important text feature is lost, residual concatenation is used to obtain the final text embeddings, this process represented as

$$T_f = Avg_{(1)}(MLP(T_4) + T_4) \quad (14)$$

where $T_f \in R^{512}$ represents our final textual feature representation.

3.3 Loss Function

In remote sensing cross-modal retrieval, bidirectional hard triplet loss [10] is often used as its loss function. Here, we still use the bidirectional hard triplet loss to constrain our model to achieve image-text alignment. We get the visual features and text features, and get the similarity matrix by the inner product of the two, and then our bidirectional hard triplet loss is expressed as

$$\mathcal{L}_R = \sum_{\hat{T}} [\alpha - S(I, T) + S(I, \hat{T})]_+ + \sum_{\hat{I}} [\alpha - S(I, T) + S(\hat{I}, T)]_+ \quad (15)$$

where α represents the marginal parameter, $S(I, T)$ is the similarity of the matched sample pairs, $S(I, \hat{T})$ and $S(\hat{I}, T)$ are the mismatched image–text pairs. The purpose of triplet loss is to make the distance between the anchor point and the unmatched sample as far as possible and to make it as close as possible to the matched sample.

We find that bidirectional hard triplet loss tends to focus only on the most difficult negative samples in each batch. However, not all samples in each batch are highly distinguishable, which is particularly prominent in remote sensing images, because there may be great similarity between different types in remote sensing images. If we only focus on the least similar negative samples, we will find that the loss of bidirectional hard triples is not enough. This tends to lose the relationship with some similar samples. According to this, we use binary cross-entropy loss (BCE loss) [16] to guide the convergence of the model in the concept space of image and text by using the entity semantic words and phrases extracted from the remote sensing text as labels for remote sensing images and text. The following is a given image-text pair (I, T) and the corresponding real label y calculation method, which is defined as follows:

$$\mathcal{L}_{\text{bce}_I}(I, y) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(g(I)_i) + (1 - y_i) \log(1 - g(I)_i)] \quad (16)$$

$$\mathcal{L}_{\text{bce}_T}(T, y) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(g(T)_i) + (1 - y_i) \log(1 - g(T)_i)] \quad (17)$$

where $g(I)$ and $g(T)$ represent visual and textual representations in the conceptual space that predict the label output at the corresponding i -th position. We constrain our model by adopting BCE loss, and the goal of the model is to minimize the difference between the predicted probability and the true label, thus aligning visual and textual patterns in the conceptual space. This loss function can help us to increase the model's attention to salient concepts, and thus improve the model's retrieval efficiency.

Finally, we get our loss function:

$$\mathcal{L} = \mathcal{L}_R + \mathcal{L}_{\text{bce}_I} + \mathcal{L}_{\text{bce}_T} \quad (18)$$

4 Experiments

4.1 DataSets and Metrics

We selected two public remote sensing image–text retrieval datasets RSICD [14] and RSITMD [23]. Each image in both datasets has five-sentence related description. RSICD contains 10,921 images, and RSITMD contains 4,743 images, which

contain more fine-grained sentences than RSICD. In the experiment, we follow the data partitioning method of [18, 23, 25]. Use 80%, 10% and 10% datasets as training sets, validation sets and test sets, respectively. In the ablation experiment, we fixed the training set and the validation set to reduce the influence of the data distribution on the model performance. We used the popular Recall@k (R@k, k = 1, 5, 10) and meanRecall (mR) [8] as evaluation metrics to evaluate the performance of sentence retrieval and image retrieval. Specifically, the goal of R@k is to calculate the proportion of the underlying truth values in the first k samples. mR Is the average of all six R@k, which accounts for the overall retrieval performance.

4.2 Implementation Details

All of the experiments in this article were conducted on a single NVIDIA RTX 3090 GPU. For images of different sizes in different datasets, we uniformly scale to 224×224 pixels and input them into the model. We performed a series of data enhancements, such as rotation and flipping, to improve the robustness of the model. The representation dimension of the word vector is set to 300. The embedding space for both image and text is 512. We use Adam [11] as the optimizer for our model. For triplet loss calculations, the margin is limited to 0.2. We use triplet loss and BCE loss, set batchsize to 100, initial learning rate to 0.0002, and decay 0.7 for every 20 epochs to train the network for 100 epochs.

Table 1. Comparison results of the cross-modal retrieval on RSICD and RSITMD.

Method	RSICD							RSITMD						
	Sentence Retrieval			Image Retrieval				Sentence Retrieval			Image Retrieval			
	R@1	R@5	R@10	R@1	R@5	R@10	mR	R@1	R@5	R@10	R@1	R@5	R@10	mR
VSE++	4.14	16.86	23.64	5.12	14.78	25.31	14.97	9.15	20.72	32.23	7.59	26.25	41.58	22.92
SCANi2t	5.11	13.56	21.76	3.70	15.12	24.89	14.02	8.94	22.26	33.41	7.45	26.14	38.68	22.81
SCANt2i	4.88	16.12	24.46	3.78	15.72	28.41	17.18	7.11	20.46	31.07	6.78	26.12	42.32	22.31
CAMP-triplet	5.12	12.67	21.56	4.20	15.38	28.10	14.50	11.77	27.09	37.89	8.52	28.01	44.25	26.25
CAMP-bce	4.23	10.54	14.86	2.88	12.57	23.05	11.35	9.12	23.14	32.89	5.43	23.27	38.24	22.01
LW-MCR	3.29	12.52	19.93	4.66	17.51	30.02	14.66	10.18	28.98	39.82	7.79	30.18	49.78	27.79
AMFMN	5.21	14.72	21.57	4.08	17.00	30.60	15.53	10.63	24.78	41.81	11.51	34.69	54.87	29.72
GaLR	6.59	19.85	31.04	4.69	19.48	32.13	18.96	14.82	31.64	42.48	11.15	36.68	51.68	31.41
KCR	5.95	18.59	29.58	5.40	22.44	37.36	19.89	-	-	-	-	-	-	-
HyperMatch	7.14	20.04	31.02	6.08	20.37	33.82	19.75	11.73	28.10	38.05	8.16	32.31	46.64	27.67
SSJDN	6.50	19.70	30.10	4.90	20.20	36.50	19.65	12.20	29.40	44.20	10.80	42.40	68.90	34.62
SIRS	5.21	20.00	30.78	5.76	20.50	33.78	19.69	15.25	35.00	48.73	11.98	41.29	60.15	35.40
SWAN	7.41	20.13	30.86	5.56	22.26	37.41	20.61	13.35	32.15	46.90	11.24	40.40	60.60	34.11
ESFN(Ours)	8.14	21.04	33.03	6.48	24.08	40.16	22.16	18.81	36.50	52.43	12.52	43.14	62.79	37.70

4.3 Comparisons With the SOTA Methods

We compare our proposed ESFN method with the latest methods on the RSICD and RSITMD datasets, which fall into two categories: Traditional cross-modal retrieval methods include visual-semantic embeddings (VSE++) [4], stacked

cross attention network (SCAN) [12], cross-modal adaptive message passing (CAMP) [20] and remote sensing cross-modal retrieval methods: asymmetric multimodal feature matching networks (AMFMN) [23], lightweight multi-scale cross-modal retrieval (LW-MCR) [24], global and local information (GaLR) [25], knowledge aware cross-modal retrieval (KCR) [17], hypergraph-enhanced textual-visual matching network (HyperMatch) [22], semantic-guided image–text retrieval framework with segmentation (SIRS) [28], scale-semantic joint decoupling network (SSJDN) [27], and scene aware aggregation networks (SWAN) [18]. The experimental results of the traditional cross-modal retrieval method were obtained by averaging three sets of repeated experiments on RSICD and RSITMD using the provided official source code. The latter directly cites the original paper for the best results.

Table 1 shows the experimental results on different datasets. We can see that it is not feasible to directly transfer the traditional cross-modal methods to remote sensing data, and the retrieval accuracy of the traditional method is far lower than that of the remote sensing method. On the RSICD and RSITMD datasets, we have seen significant improvements in our metrics. On RSICD, our R@1 and mR Scores reached 8.14 and 22.16 respectively. They are 9.8% and 7.5% higher than SWAN model respectively. Compared with RSICD, RSITMD contains more fine-grained entities. On RSITMD, the scores of R@1 and mR Reach 18.81 and 37.70 respectively, which are 40.8% and 10.5% higher than those of SWAN model. The performance on these two datasets illustrates the validity of our model.

4.4 Ablation Studies of Structures

In this section, we further analyze our ESFN model, and we design six controlled experiments to further analyze the effectiveness of our proposed module.

Table 2. Ablation experiment for different structures of ESFN on RSITMD testset

Ablation model	Img.F		Text.F		Loss		Sentence Retrieval			Image Retrieval			mR	
	SEF	MAF	High	LSTM	TPE	Triplet	BCE	R@1	R@5	R@10	R@1	R@5	R@10	
base			✓	✓		✓		11.57	26.11	42.26	11.15	37.57	60.66	31.55
L.1	✓		✓	✓		✓		13.72	30.97	44.47	11.90	38.27	61.28	32.98
L.2		✓	✓	✓		✓		15.93	31.64	46.46	11.50	40.93	61.68	34.69
L.3	✓	✓	✓	✓		✓		16.37	34.51	48.23	12.79	41.28	62.08	35.88
T.1	✓	✓	✓	✓	✓	✓		16.15	37.17	48.01	12.08	42.39	63.45	36.54
full	✓	✓	✓	✓	✓	✓	✓	18.81	36.50	52.43	12.52	43.14	62.79	37.70

- base: Only High-level features extracted by CNN are used in the visual extraction part, only one-way LSTM is used in the textual extraction part, entity semantic keywords in the text are embedded in the memory unit in the LSTM.
- L.1: Uses SEF module and High-level features extracted by CNN in the visual extraction section, be consistent with base in the textual extraction section.

- I_2: Uses MAF module and High-level features extracted by CNN in the visual extraction section, be consistent with base in the textual extraction section.
- I_3: Uses SEF module, MAF module and High-level features extracted by CNN in the visual extraction section, be consistent with base in the textual extraction section.
- T_1: Uses SEF module, MAF module and High-level features from CNN in the visual extraction part, use only one-way LSTM in the textual extraction part, and embed entity keywords in the LSTM memory unit, and add TPE module to enhance the focus on entity words and phrases.
- full: Uses SEF module, MAF module and High-level features extracted by CNN in the visual extraction part, use only one-way LSTM in the textual extraction part, and embed entity keywords in the text into the memory unit of LSTM, and add TPE module to enhance the focus on entity words and phrases. Add a BCE loss constraint.

Table 2 shows the results of the five experiments. Compared with the base model, by adding SEF, the mR of I_1 is increased by 4.5%, and the ratio of Image Retrieval and Sentence Retrieval R@1 respectively increased by 18.5% and 6.7%. With the increase of SEF, the retrieval precision of R@1 in the visual part is greatly improved. When MAF is added, the mR of I_2 is increased by 9.9%. In the visual part, all the visual modules are added. Compared with the base model, mR is improved by 13.7%, compared with the I_1 model, mR is improved by 8.1%, compared with the I_2 model, mR is improved by 3.4%. The I_3 model makes full use of the features extracted from each layer of ResNet and filters the entity semantic feature of low-level features, mid-level feature adaptive selection fusion and high-level feature filtering get good results. After the addition of TPE module, the attention to text entity semantic feature in text is further improved, and mR is improved by 1.8% compared with I_3 model. After finally adding BCE loss, the model has further constraints on the learning of entity concepts, so as to achieve the best retrieval effect.

4.5 The Validity of Entity Word Embeddedness

In this section, we conduct a series of experiments that analyze of the effectiveness of embedding eighty words and phrases into the memory unit in LSTM. We select four models with different structures to verify the validity of entity words embeddings. The visual extraction part of the following model is consistent with the full ESFN model, the text extraction section adds TPE module and BCE loss is added to further constrain the model.

Table 3. Performance comparison of whether entity words are embedded on the RSITMD testset

Ablation model	Sentence Retrieval			Image Retrieval			mR
	R@1	R@5	R@10	R@1	R@5	R@10	
LSTM no KEY	13.94	36.06	51.55	12.52	42.92	64.12	36.85
BI-LSTM no KEY	16.37	37.17	50.00	12.70	43.23	62.70	37.03
LSTM with KEY	18.81	36.50	52.43	12.52	43.14	62.79	37.70
BI-LSTM with KEY	13.72	35.62	50.44	13.98	43.27	63.81	36.81

- uses one-way LSTM without embedding entity words.
- uses bidirectional LSTM without embedding entity words.
- uses one-way LSTM to embed entity words.
- uses bidirectional LSTM to enable the embedding of entity words.

It can be seen from the data in the Table 3 that, when entity words are not embedded, bidirectional LSTM achieves better retrieval results than unidirectional LSTM. Bidirectional LSTM learns text semantics from two different directions, and has stronger context understanding ability compared with unidirectional LSTM. However, after the entity words are embedded, the retrieval effect changes. The effect of bidirectional LSTM embedded entity words is not as good as that of unidirectional LSTM embedded entity words. When we use bidirectional LSTM embedded entity words, the embedding order of entity words is reversed accordingly. To some extent, the two texts with different order and the reversed word sequence interfered with the LSTM learning, resulting in the decline of the retrieval effect. From the experimental results of whether to embed entity words in LSTM, it can be seen that embedding entity words can effectively improve the performance of image retrieval.

4.6 Visualization and Analysis

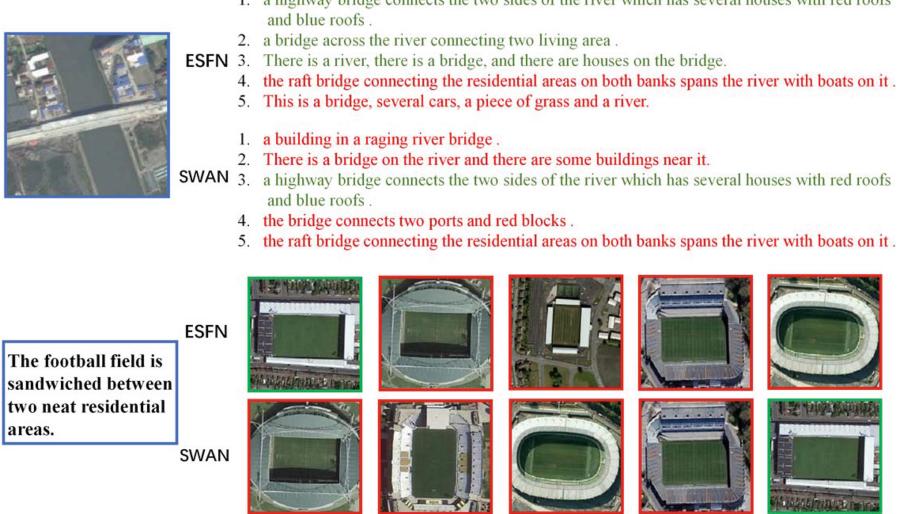


Fig. 4. Visual results of bidirectional retrieval of RSITMD dataset. Green fonts and boxes indicate correct search results, and red fonts and boxes indicate incorrect search results (Color figure online)

In the sentence retrieval results illustrated in Fig. 4, our model demonstrates superior performance compared to other methods. Specifically, to provide a more intuitive reflection of our model’s retrieval efficacy, we show a comparison with the SWAN model’s retrieval performance. In text retrieval tasks, our ESFN model retrieves three corresponding image descriptions successfully outperforming the SWAN model [18]. Moreover, our ESFN model places greater emphasis on capturing entity semantic present in images, such as bridges, red roofs, blue roofs, and other vital entities. In terms of image retrieval, our ESFN model retrieves the target images corresponding to the text at the top rank, while the SWAN model retrieves the target image but with a lower ranking. In the example in the second row, both the ESFN model and the SWAN model identify similar image samples, i.e., a soccer field. However, remote sensing images often exhibit significant similarities, and our ESFN model excels at capturing entity semantic from the text—“sandwiched between two tidy residential areas.” This capability contributes to our superior retrieval performance. Therefore, this search logic is deemed reasonable.

5 Conclusion

In this paper, we propose the ESFN network to improve the retrieval effect by strengthening the entity semantic in remote sensing images and texts. We

propose the SEF Module to extract entity semantic feature of different sizes contained in remote sensing images. At the same time, we propose the MAF module to adaptively fusion multi-scale visual features to enhance visual representation. By aggregating the low-level, mid-level and high-level information of the image, the visual representation containing rich entity semantic features is obtained. In addition, to align visual modality, we embed the entity semantic in the text into our textual feature extractor and propose a TPE module to enhance text entity semantic representation. A large number of experiments have been performed on RSICD and RSITMD datasets, and the results show the superiority of this method.

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

References

1. Abdullah, T., Bazi, Y., Al Rahhal, M.M., Mekhalfi, M.L., Rangarajan, L., Zuair, M.: TextRS: deep bidirectional triplet network for matching text to remote sensing images. *Remote Sens.* **12**(3), 405 (2020)
2. Cheng, Q., Zhou, Y., Fu, P., Xu, Y., Zhang, L.: A deep semantic alignment network for the cross-modal image-text retrieval in remote sensing. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **14**, 4284–4297 (2021)
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. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017)
5. Feng, D., He, X., Peng, Y.: MKVSE: multimodal knowledge enhanced visual-semantic embedding for image-text retrieval. *ACM Trans. Multimed. Comput. Commun. Appl.* **19**(5), 1–21 (2023)
6. 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)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
8. Huang, Y., Wu, Q., Song, C., Wang, L.: Learning semantic concepts and order for image and sentence matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6163–6171 (2018)
9. Ji, Z., Meng, C., Zhang, Y., Pang, Y., Li, X.: Knowledge-aided momentum contrastive learning for remote-sensing image text retrieval. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–13 (2023)
10. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
12. Lee, K.-H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV*

2018. LNCS, vol. 11208, pp. 212–228. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_13
- 13. Liu, S., Huang, D., Wang, Y.: Learning spatial fusion for single-shot object detection. arxiv 2019. arXiv preprint [arXiv:1911.09516](https://arxiv.org/abs/1911.09516) (1911)
 - 14. Lu, X., Wang, B., Zheng, X., Li, X.: Exploring models and data for remote sensing image caption generation. *IEEE Trans. Geosci. Remote Sens.* **56**(4), 2183–2195 (2017)
 - 15. Lv, Y., Xiong, W., Zhang, X., Cui, Y.: Fusion-based correlation learning model for cross-modal remote sensing image retrieval. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2021)
 - 16. Mannor, S., Peleg, D., Rubinstein, R.: The cross entropy method for classification. In: *Proceedings of the 22nd International Conference on Machine Learning*, pp. 561–568 (2005)
 - 17. Mi, L., Li, S., Chappuis, C., Tuia, D.: Knowledge-aware cross-modal text-image retrieval for remote sensing images. In: *Proceedings of the Second Workshop on Complex Data Challenges in Earth Observation (CDCEO 2022)* (2022)
 - 18. Pan, J., Ma, Q., Bai, C.: Reducing semantic confusion: Scene-aware aggregation network for remote sensing cross-modal retrieval. In: *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, pp. 398–406 (2023)
 - 19. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
 - 20. Wang, Z., et al.: CAMP: cross-modal adaptive message passing for text-image retrieval. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5764–5773 (2019)
 - 21. Xia, G.S., et al.: AID: a benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **55**(7), 3965–3981 (2017)
 - 22. Yao, F., et al.: Hypergraph-enhanced textual-visual matching network for cross-modal remote sensing image retrieval via dynamic hypergraph learning. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* **16**, 688–701 (2022)
 - 23. Yuan, Z., et al.: Exploring a fine-grained multiscale method for cross-modal remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–19 (2022)
 - 24. Yuan, Z., et al.: A lightweight multi-scale crossmodal text-image retrieval method in remote sensing. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–19 (2021)
 - 25. Yuan, Z., et al.: Remote sensing cross-modal text-image retrieval based on global and local information. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–16 (2022)
 - 26. Zeng, Y., Zhang, X., Li, H., Wang, J., Zhang, J., Zhou, W.: X2-VLM: all-in-one pre-trained model for vision-language tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023)
 - 27. Zheng, C., Song, N., Zhang, R., Huang, L., Wei, Z., Nie, J.: Scale-semantic joint decoupling network for image-text retrieval in remote sensing. *ACM Trans. Multimed. Comput. Commun. Appl.* **20**(1), 1–20 (2023)
 - 28. Zhu, Z., Kang, J., Diao, W., Feng, Y., Li, J., Ni, J.: SIRS: multi-task joint learning for remote sensing foreground-entity image-text retrieval. *IEEE Trans. Geosci. Remote Sens.* (2024)



Semantic Preservation and Hash Fusion Network for Unsupervised Cross-Modal Retrieval

Xinsheng Shu and Mingyong Li^(✉)

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

Abstract. With the exponential growth of multimedia content on the Internet, cross-modal retrieval has emerged as a critical research area. This task aims to efficiently and accurately retrieve relevant information across different modalities based on a given query. Despite some unsupervised cross-modal hashing methods proposed for image-text retrieval with unlabeled data, existing methods struggle with extracting semantic features, preserving multi-modal semantics, and ensuring modal interaction. To address these issues, we proposed the Semantic Preservation and Hash Fusion Network (SPHFN). Our approach includes a Long-Range Semantic Capturing module to capture extensive dependencies in text and an Identity Semantic Preservation module to maintain intrinsic semantic information of original samples. These representations are then combined to ensure semantic consistency across modalities. We also construct a joint similarity matrix to generate high-quality unified binary hash codes by leveraging the interaction among continuous codes from different modalities. Experimental results on multiple datasets show that our method significantly outperforms existing state-of-the-art approaches in cross-modal hashing retrieval tasks.

Keywords: cross-modal retrieval · hashing · semantic preservation · hash fusion

1 Introduction

In today's mobile internet era, the rapid growth of multi-modal data, such as image, text, audio, and video, has made cross-modal retrieval a highly researched field. This task has significant practical implications and can be applied in various fields including multimedia retrieval, intelligent recommendation systems, and visual voice assistants.

Over the past few years, some deep unsupervised cross-modal hashing methods have demonstrated impressive performance based on deep learning, such as JDSH [16], DJSRH [21], AGCH [32], and CIRH [33]. However, cross-modal

retrieval still faces several challenges. Firstly, there exist differences in data representations between different modalities, such as image and text, leading to a semantic gap between modalities. Different modalities have different ways of representing features, for example, image can be represented by pixel values, while text can be represented by word vectors. So, a key challenge lies in capturing common information and complementary features between modalities through effective cross-modal feature representation learning, and how to match and fuse these different representations to enable effective cross-modal learning. Secondly, previous methods have primarily focused on feature fusion. They typically construct modality-specific similarity matrices separately and then combine these matrices based on image-text matching information to obtain the final instance similarity matrix. This approach often lacks modality interactions or has weak semantic interactions between different modalities when constructing instance similarity matrices. As a result, it can lead to incomplete or inaccurate cross-modal associations. Examples of such methods include JDSH [16], DSAH [27] and DGCPN [28]. Therefore, another challenge is to improve interactions between different modal features and hash codes. It is important to assign different weights to each modality rather than treating them equally. This requires better interaction and differentiation between the modalities.

To address these challenges, one such model that has gained significant attention is CLIP. The CLIP [18] has showcased impressive semantic understanding capabilities and remarkable zero-shot or few-shot learning abilities. CLIP's success has greatly transformed multiple fields including the field of cross-modal retrieval. JM-CLIP [6], CCAH [14], CAGAN [15] utilizing CLIP for cross-modal retrieval have shown significant performance improvements, the exploration of how CLIP knowledge benefits 3D scene understanding in Clip2Scene [1], and the application of the CLIP model for scene text detection in TurningA-ClipModel [29]. The strength of CLIP lies in its ability to jointly learn image and text representations, achieving a unified representation learning between image and text. However, CLIP's performance in the field of cross-modal retrieval is limited by the length requirements of its text encoder.

In this paper, we conducted extensive research to explore how to fully leverage deep semantic information from text and images to improve retrieval performance. The main contributions of our approach can be summarized as follows:

1. We proposed a method that effectively captures long-distance dependencies in text, facilitating the extraction of rich feature representations in both text and images. Subsequently, we thoroughly integrate these representations to enhance semantic consistency between cross-modal and within-modal data.
2. We introduced a novel approach that combines a Long-CLIP-based Binary Code Fusion and Selection module with an Identity Semantic Preservation module. This joint framework enhances the quality of the generated hash codes while preserving intrinsic semantic information. By leveraging a joint similarity matrix, our method ensures that the learned feature representations retain their semantic integrity, thus contributing to both improved hash code generation and semantic preservation.

3. Experimental evaluations on multiple publicly available benchmark datasets demonstrate that our method surpasses state-of-the-art unsupervised approaches in cross-modal retrieval.

2 Related Work

2.1 Unsupervised Cross-Modal Hashing

Unsupervised cross-modal retrieval aims to learn semantic correlations from unlabeled multi-modal data, with the goal of connecting semantically similar data from different modalities to enable effective cross-modal retrieval. Existing methods, including deep learning-based models, utilize structures such as autoencoders, generative adversarial networks (GANs), or Variational Autoencoders (VAEs) for cross-modal representation learning. These approaches map different modalities to a common low-dimensional space where data with similar semantics are closer in distance. JDSH [16] proposes a distribution-based similarity decision and weighting approach to generate more discriminative hash codes. CIRH [33] designs a multi-modal collaborated graph to construct heterogeneous multi-modal correlations and performs semantic aggregation on graph networks to generate a multi-modal complementary representation. UGACH [31] leverages the unsupervised representation learning capability of GANs to exploit the underlying manifold structure of cross-modal data. DJSRH [21] constructs a unified semantic affinity matrix from the neighborhood structures of original data which is then reconstructed to generate hash codes. AGCH [32] constructs a similarity matrix by aggregating multi-modal similarity matrices generated by different similarity measurement methods and uses graph convolutional networks (GCN) to preserve the semantic structure in the hash codes.

2.2 Contrastive Learning-Based Cross-Modal Retrieval

The core concept of contrastive learning involves categorizing samples into positive and negative pairs. The model is then trained to maximize the similarity of positive pairs while minimizing the similarity of negative pairs. This approach aims to reinforce the model's ability to distinguish between similar and dissimilar samples, improving its overall performance. SimCLR [2] proposes a simple yet efficient framework for contrastive learning and discovers some regularities in unsupervised contrastive learning. MoCo (Momentum Contrast) [8] improves the efficiency of contrastive learning by introducing a queue to store feature representations. CLIP [18] has also received widespread attention. DUMCH [17] introduces momentum contrastive learning for unsupervised cross-modal hashing to flexibly define a robust loss by comparing positive and negative samples. ELR-CMR [26] introduces early learning regularization to prevent memorizing noisy labels and utilizes intra-modal contrastive learning for robust feature embedding.

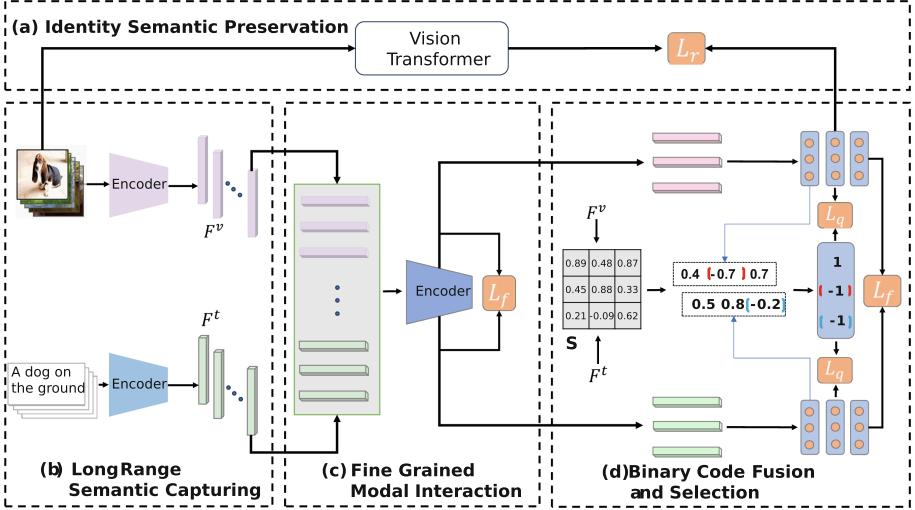


Fig. 1. We utilize modules (a) and (b) to extract features from the original images. Subsequently, the features extracted by module (b) from both text and images are fed into module (c) for fine-grained semantic interactions. Finally, in module (d), we employ the features extracted by module (b) to guide our hash code generation. Meanwhile, during the process of generating hash codes, we reconstruct the features extracted by module (a) to preserve the original semantics

3 Methodology

3.1 Framework Overview

The network architecture of our proposed method is illustrated in Fig. 1. It comprises four main modules: (a) Identity Semantic Preservation, (b) Long-Range Semantic Capturing, (c) Fine-Grained Multi-modal Contrastive Learning, and (d) Long-CLIP-based Binary Code Fusion and Selection. The Long-Range Semantic Capturing module employs Long-CLIP [30] to capture long-distance dependency relationships in text. The Fine-Grained Multi-modal Contrastive Learning module utilizes a multi-modal fusion Transformer encoder to facilitate fine-grained semantic interactions among the extracted features. The Identity Semantic Preservation module uses an Encoder to reconstruct the identity information of the images. The Long-CLIP-based Binary Code Fusion and Selection module efficiently generates a joint similarity matrix to guide the generation of high-quality hash codes. In the following sections, we will provide a detailed description of the proposed method.

3.2 Notation and Problem Definition

To facilitate the introduction of our methods, let's define some symbols. In our paper, uppercase bold letters, e.g., \mathbf{A} , and lowercase bold letters, e.g., \mathbf{a} , represent matrices and vectors, respectively. Assuming we have a training set $o = \{x_i, y_i\}_{i=1}^n$, where x_i and y_i represent the visual and text data of the i^{th} sample, $\mathbf{F}^v = \{\mathbf{f}_i^v\}_{i=1}^n$ denotes the extracted image features, and $\mathbf{F}^t = \{\mathbf{f}_i^t\}_{i=1}^n$ denotes the extracted text features. Here, $\mathbf{f}_i^v \in \mathbf{R}^{1 \times d_v}$ and $\mathbf{f}_i^t \in \mathbf{R}^{1 \times d_t}$ represent the visual and text feature extracted from the i^{th} sample's original data. n , d_v , d_t , v , and t denote the number of samples, the dimension of visual features, the dimension of text features, image modal, and text modal, respectively. $\mathbf{H}^v = \{\mathbf{h}_i^v\}_{i=1}^n \in [-1, 1]^{k \times n}$ and $\mathbf{H}^t = \{\mathbf{h}_i^t\}_{i=1}^n \in [-1, 1]^{k \times n}$ represent the continuous hash codes generated by our hash generation networks for images and text, respectively. Where k is the length of the hash codes. Our ultimate goal is to learn a compact binary representation $\mathbf{b}_i \in \{-1, 1\}^{1 \times k}$ for each sample. Thus, the hash codes matrix can be defined as $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^n$.

3.3 Long-Range Semantic Capturing

CLIP achieves deep semantic understanding of both image and text. It not only extends the model's ability to leverage large-scale data but also provides a versatile multi-modal model that has achieved remarkable performance across various tasks.

Although CLIP has demonstrated state-of-the-art performance on various multi-modal tasks, it has a notable limitation when it comes to text input length. The length of text prompts of CLIP's text encoder is limited to 77 tokens, and empirical studies [30] have shown that the effective length is even smaller, possibly below 20 tokens. The absence of the long-text capability not only restricts the potential, to capture details and relationships within images and texts, of the text encoder and the image encoder of CLIP but also greatly reduces its performance in cross-modal retrieval. Benchmark datasets commonly used in cross-modal retrieval tasks, such as MIRFlickr-25K [10], NUS-WIDE [3], and IAPR TC-12 [5], and so on, contain samples with text lengths that exceed 77 tokens, severely limiting CLIP's performance in cross-modal hashing retrieval tasks.

A potential approach [30] to address this issue involves relaxing the strict constraints on input text length by incorporating positional embedding interpolation, followed by fine-tuned CLIP using image-text pairs containing long descriptions. To fully explore the capabilities of CLIP in cross-modal retrieval with its long-text capability unlocked, we utilize the fine-tuned encoder of Long-CLIP to capture rich textual feature representations \mathbf{F}^t and image feature representations \mathbf{F}^v , for one sample:

$$\mathbf{f}_i^t = \text{LongEncoder}_{txt}(y_i), \mathbf{f}_i^v = \text{LongEncoder}_{img}(x_i) \quad (1)$$

3.4 Fine-Grained Multi-modal Contrastive Learning

As described in UCMFH [25], to enable comprehensive fine-grained semantic interaction between images and texts, we utilize a standard Transformer Encoder [24] architecture to model the correspondence between image-text pairs. To achieve this, we first concatenate the outputs of the feature encoders along the first dimension and then feed them into the Transformer Encoder for thorough fine-grained semantic interaction between images and text. The formal representation is as follows:

$$\mathbf{z} = \text{Transformer-Encoder}(\text{concatenate}(\mathbf{f}_i^v, \mathbf{f}_i^t; \text{dim} = 1)), \quad (2)$$

where $\mathbf{z} = \text{concatenate}[\mathbf{z}^v \mathbf{z}^t]$, \mathbf{z}^v and \mathbf{z}^t represent the image and text feature vectors through semantic interaction with each modality, respectively. Next, we introduce a hash network for each modality to learn continuous hash codes:

$$\mathbf{b}_i^v = \text{sgn}(\mathcal{MLP}(\mathbf{z}_i^v; \theta_v)), \quad \mathbf{b}_i^t = \text{sgn}(\mathcal{MLP}(\mathbf{z}_i^t; \theta_t)), \quad (3)$$

where θ_v and θ_t represent the parameters of the MLP networks for images and texts, respectively. $\text{sgn}(\cdot)$ denotes the sign function. To ensure that data representations of the same category in different modalities contain consistent category semantics, we define inter-modal contrastive loss as follows:

$$\begin{aligned} \mathcal{L}_{fusion_inter}^v &= - \sum_{i=1}^m \log \frac{\exp(\cos(\mathbf{p}_i^v, \mathbf{p}_i^{v+})/\tau)}{\sum_{j=1}^m \exp(\cos(\mathbf{p}_i^v, \mathbf{p}_i^t)/\tau)}, \\ \mathcal{L}_{fusion_inter}^t &= - \sum_{i=1}^m \log \frac{\exp(\cos(\mathbf{p}_i^t, \mathbf{p}_i^{t+})/\tau)}{\sum_{j=1}^m \exp(\cos(\mathbf{p}_i^t, \mathbf{p}_i^v)/\tau)}, \end{aligned} \quad (4)$$

To further narrow the semantic gap within the modal, we also define the intra-modal contrastive loss:

$$\begin{aligned} \mathcal{L}_{fusion_inner}^v &= - \sum_{i=1}^m \log \frac{\exp(\cos(\mathbf{p}_i^v, \mathbf{p}_i^{v+})/\tau)}{\sum_{j=1}^m \exp(\cos(\mathbf{p}_i^v, \mathbf{p}_i^v)/\tau)}, \\ \mathcal{L}_{fusion_inner}^t &= - \sum_{i=1}^m \log \frac{\exp(\cos(\mathbf{p}_i^t, \mathbf{p}_i^{t+})/\tau)}{\sum_{j=1}^m \exp(\cos(\mathbf{p}_i^t, \mathbf{p}_i^t)/\tau)}, \end{aligned} \quad (5)$$

where m represents the batch size, τ denotes temperature coefficient, \cos represents cosine similarity function, $\mathbf{p} \in \{\mathbf{z}, \mathbf{b}\}$. $\mathbf{p}_i^v, \mathbf{p}_i^{v+}$ and $\mathbf{p}_i^t, \mathbf{p}_i^{t+}$ represent the representations of the truly aligned image-text pairs, and so do $\mathbf{p}_i^v, \mathbf{p}_i^{v+}$ and $\mathbf{p}_i^t, \mathbf{p}_i^{t+}$. Thus, the modality contrastive loss can be defined as follows:

$$\mathcal{L}_f = \mathcal{L}_{fusion_inter}^v + \mathcal{L}_{fusion_inter}^t + \mathcal{L}_{fusion_inner}^v + \mathcal{L}_{fusion_inner}^t. \quad (6)$$

3.5 Long-CLIP-Based Binary Code Fusion and Selection

We utilize MLP networks to generate continuous hash codes \mathbf{H}^v and \mathbf{H}^t . Subsequently, applying the $\text{sgn}(\cdot)$ function yields their corresponding binary hash

codes $\mathbf{B}^v = \text{sgn}(\mathbf{H}^v) = \{\mathbf{b}_i^v\}_{i=1}^n \in \{-1, 1\}^{k \times n}$ and $\mathbf{B}^t = \text{sgn}(\mathbf{H}^t) = \{\mathbf{b}_i^t\}_{i=1}^n \in \{-1, 1\}^{k \times n}$. The binary hash codes \mathbf{b}_i^v and \mathbf{b}_i^t for an image x_i and its corresponding text y_i are expected to be identical since they stem from the same instance o_i with equivalent semantic information. In line with conventional methods [9], if $\mathbf{h}_i^v = (0.4, -0.7, 0.7)$ and $\mathbf{h}_i^t = (0.5, 0.8, -0.2)$, quantized binary hash codes \mathbf{c}_i^v and \mathbf{c}_i^t are generated as $\text{sgn}(\mathbf{h}_i^v + \mathbf{h}_i^t)$. This would result in a quantized binary code \mathbf{c}_i of $(1, 1, 1)$. However, if the quality of the second bit in the image hash code exceeds that of the corresponding bit in the text hash code, and conversely, if the quality of the third bit in the text hash code surpasses that of the corresponding bit in the image hash code, the preferable value for \mathbf{c}_i would be $(1, -1, -1)$. Given CLIP's remarkable performance in various downstream tasks and its capacity to extract comprehensive image and text feature representations, we utilize the Encoder of Long-CLIP. This enhanced version of CLIP is equipped to handle long texts efficiently, enabling us to extract text and image feature representations. Subsequently, we construct a similarity matrix to evaluate the quality of each bit in the hash codes. The formal representation of our similarity matrix \mathbf{S} is as follows:

$$\mathbf{S} = \cos(\mathbf{F}^v, \mathbf{F}^t), \quad (7)$$

Inspired by UCHM [22], we define our objective function \mathcal{L} :

$$\mathcal{L} = \|\cos(\mathbf{B}, \mathbf{B}) - \mathbf{S}\|_F^2 = \sum_{ij} \left(\frac{1}{k} \mathbf{b}_i^\top \mathbf{b}_j - \mathbf{S}_{ij} \right)^2, \quad (8)$$

where \mathbf{B} represents binary hash codes, \mathbf{S}_{ij} represents the i^{th} row and j^{th} column of \mathbf{S} . The objective function can be further optimized as follows:

$$\min_{\Lambda} \mathcal{L} = \sum_{ij} \left(\frac{1}{k} \sum_{z=1}^k \lambda_z \mathbf{b}_{iz}^\top \mathbf{b}_{jz} - \mathbf{S}_{ij} \right)^2 = \left\| \frac{1}{k} \mathbf{B}^\top \Lambda \mathbf{B} - \mathbf{S} \right\|_F^2, \quad (9)$$

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_z \end{pmatrix},$$

Given fixed values of \mathbf{B} and \mathbf{S} , λ_z can be used to represent the quality of the z^{th} bit of the hash code, with a higher value indicating a higher quality of the z^{th} bit of the hash code. Next, we define our cross-modal joint hash code matrix \mathbf{D} as \mathbf{B} in the above equation:

$$\mathbf{D} = \text{sgn} \left(\begin{bmatrix} \mathbf{H}^v \\ \mathbf{H}^t \end{bmatrix} \right) \in \{-1, 1\}^{2k \times n}, \quad (10)$$

Then, the vector $\boldsymbol{\lambda}$ of the diagonal elements of Λ can be calculated:

$$\boldsymbol{\lambda} = k[(\mathbf{D}\mathbf{D}^\top) \circ (\mathbf{D}\mathbf{D}^\top)]^{-1} \text{diag}(\mathbf{D}\mathbf{S}\mathbf{D}^\top), \boldsymbol{\lambda} \in \mathcal{R}^{2k}, \quad (11)$$

where $diag(\cdot)$ is used to extract the diagonal elements of a matrix. Finally generate our bit selection vector \mathbf{r} :

$$\mathbf{r} = \begin{cases} 1, \text{if } \lambda_i > \lambda_{i+k} \\ 0, \text{otherwise} \end{cases} \in \mathcal{R}^k, \quad (12)$$

where $i \in \{1, 2, \dots, k\}$. When r_i equals 1, it indicates that the quality of the i -bit in the image hash code surpasses that of the text hash code for the same bit position. And generate the unified hash codes \mathbf{C} :

$$\mathbf{C}^v = \mathbf{C}^t = \mathbf{C} = (\mathbf{r} \mathbf{1}^\top) \circ \text{sgn}(\mathbf{H}^v) + (1 - \mathbf{r} \mathbf{1}^\top) \circ \text{sgn}(\mathbf{H}^t), \quad (13)$$

where $\mathbf{1}$ is a $1 \times n$ dimensional matrix with element 1. In this way, a better, unified binary hash code $\mathbf{C}^v = \mathbf{C}^t = \mathbf{C}$ is generated, which further improves the retrieval performance of our hash model.

Finally, we define the quantized objective function:

$$\mathcal{L}_q = \|\mathbf{C} - \mathbf{H}^v\|_F^2 + \|\mathbf{C} - \mathbf{H}^t\|_F^2 \quad (14)$$

3.6 Identity Semantic Preservation

To capture long-range dependencies in the image and prevent the network from being over-fitted to the limited semantics of training data, the rich semantics $\mathbf{F}^{v'}$ from the pre-trained model with the pre-trained label are persevered (i.e., the outputs of the last layer in the pre-trained Vision Transformers [4]). Compared to traditional convolutional neural networks, it divides the image into patches and then achieves global perception by utilizing self-attention mechanisms, allowing it to capture long-range dependencies in images. Simultaneously, we extract intermediate layer 1000-dimension features $\mathbf{H}^{v'}$ from the image MLP network, a network with three fully-connected layers (i.e., $z^v \rightarrow 4096 \rightarrow 1000 \rightarrow h^v$), to construct a consistency reconstruction loss. The formal representation is as follows:

$$\mathcal{L}_r = \left\| \mathbf{F}^{v'} - \mathbf{H}^{v'} \right\|_F^2, \quad (15)$$

3.7 Objective Function

Incorporating Eq. 6, Eq. 14 and Eq. 15, the overall objective function is:

$$\mathcal{L} = \mathcal{L}_f + \alpha \mathcal{L}_q + \beta \mathcal{L}_r,$$

where α and β are hyper-parameters that are used to adjust the weights of loss items.

Table 1. Comparison results of MAP@5000 on MIRFlickr-25K and IAPR TC-12 datasets.

Task	Method	MIRFlickr-25K				IAPR TC-12			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
I → T	DCMH [11]	0.7376	0.7466	0.7469	0.7547	0.4526	0.4732	0.4844	0.4983
	SSAH [12]	0.7654	0.7728	0.7809	0.7892	0.5381	0.5668	0.5862	0.5992
	AGAH [7]	0.7135	0.7047	0.7154	0.7219	0.4655	0.4756	0.4813	0.4921
	JDSH [16]	0.7245	0.7342	0.7424	0.7451	0.4786	0.4927	0.5155	0.5339
	UKD [9]	0.7138	0.7184	0.7255	0.7201	0.4812	0.4920	0.4941	0.5027
	DCHUC ₂₀₂₂ [23]	0.7576	0.7557	0.7611	0.7495	0.5545	0.6023	0.6257	0.6484
	DAEH ₂₀₂₂ [19]	0.7802	0.7916	0.7984	0.8027	0.5428	0.5796	0.6011	0.6087
	SKDCH ₂₀₂₃ [20]	0.7216	0.7335	0.7382	0.7403	0.5421	0.5714	0.5998	0.6103
	CAGAN ₂₀₂₃ [15]	0.7902	0.8195	0.8248	0.8341	0.5584	0.5931	0.6272	0.6468
	UCMFH ₂₀₂₃ [25]	0.7599	0.8068	0.8434	0.8492	0.5598	0.6224	0.6500	0.6877
	CKDH ₂₀₂₄ [13]	0.8376	0.8590	0.8819	0.8818	0.5885	0.6163	0.6375	0.6570
	Ours	0.8753	0.9092	0.9126	0.9151	0.6678	0.6922	0.7125	0.7097
T → I	DCMH [11]	0.7628	0.7733	0.7792	0.7857	0.5185	0.5378	0.5468	0.5596
	SSAH [12]	0.7768	0.7847	0.7811	0.7823	0.5392	0.5646	0.5873	0.5998
	AGAH [7]	0.7039	0.6682	0.6676	0.6775	0.5193	0.5316	0.5437	0.5534
	JDSH [16]	0.7109	0.7207	0.7334	0.7254	0.4774	0.4915	0.5138	0.5332
	UKD [9]	0.7156	0.7166	0.7214	0.7190	0.4886	0.5017	0.5145	0.5172
	DCHUC ₂₀₂₂ [23]	0.7680	0.7791	0.7753	0.7681	0.5393	0.6039	0.6431	0.6734
	DAEH ₂₀₂₂ [19]	0.7597	0.7667	0.7741	0.7813	0.5132	0.5558	0.5799	0.5905
	SKDCH ₂₀₂₃ [20]	0.7814	0.7985	0.8067	0.7982	0.5404	0.6021	0.6398	0.6701
	CAGAN ₂₀₂₃ [15]	0.7790	0.8018	0.8160	0.8272	0.5541	0.6114	0.6444	0.6745
	UCMFH ₂₀₂₃ [25]	0.7640	0.8157	0.8460	0.8506	0.5619	0.6247	0.6526	0.6879
	CKDH ₂₀₂₄ [13]	0.8311	0.8598	0.8813	0.8815	0.5917	0.6239	0.6612	0.6807
	Ours	0.8759	0.9107	0.9080	0.9134	0.6658	0.6933	0.7105	0.7096

4 Experiments

4.1 Datasets

We used three widely used datasets in the field of cross-modal retrieval, and below is a brief introduction to each of them.

MIRflickr-25K [10]: This dataset contains 25,000 image-text pairs across 24 classes. We filtered for pairs with over 20 labels, resulting in 20,015 pairs. Among them, 2,000 pairs were randomly selected as the query set, and the remaining 18,015 pairs formed the retrieval set. Additionally, we chose 5,000 pairs from the retrieval set for the training set.

IAPR TC-12 [5]: This dataset comprises 20,000 image-text pairs annotated with 255 labels. For our experiment, we randomly chose 2,000 pairs as the query set, while the remaining 18,000 pairs formed the retrieval set. From the retrieval set, we selected 10,000 pairs as the training set.

NUS-WIDE [3]: This dataset contains 269,648 images across 81 classes. We created an experimental dataset by selecting 186,577 image-text pairs from the top 10 most common classes. Among them, 2,000 pairs were randomly chosen as the query set, while the remaining 184,577 pairs formed the retrieval set. From the retrieval set, we selected 5,000 pairs for the training set.

4.2 Implementation Details and Results

We compared our approach with several typical and widely used baseline methods shown in Table 1 and Table 2. All experiments were conducted on a Windows 10 system with an i5-12490F CPU and NVIDIA GeForce RTX 3090 GPU. The Python version used was 3.9.18, the PyTorch version was 2.1.0, and the CUDA version was 12.1. The number of iterations set for the experiments was 300. For NUS-WIDE, we set the parameters $\alpha = 15$ and $\beta = 1$. For MIRFlickr-25K, we set the parameters $\alpha = 10$ and $\beta = 1$. For the IAPR TC-12 dataset, we set the parameters $\alpha = 0.1$ and $\beta = 0.1$.

Table 2. Comparison results of MAP@5000 on NUS-WIDE dataset.

Task	Method	NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits
I → T	DCMH [11]	0.5783	0.5921	0.5984	0.6097
	SSAH [12]	0.6036	0.6187	0.6404	0.6428
	AGAH [7]	0.4954	0.5281	0.5923	0.5539
	JDSH [16]	0.6781	0.7248	0.7434	0.7565
	UKD [9]	0.6149	0.6378	0.6386	0.6454
	DCHUC ₂₀₂₂ [23]	0.7501	0.7759	0.7976	0.7953
	DAEH ₂₀₂₂ [19]	0.7299	0.7534	0.7708	0.7769
	SKDCH ₂₀₂₃ [20]	0.7314	0.7444	0.7642	0.7701
	CAGAN ₂₀₂₃ [15]	0.7565	0.7890	0.7950	0.8059
	UCMFH ₂₀₂₃ [25]	0.7867	0.8280	0.8270	0.8533
	CKDH ₂₀₂₄ [13]	0.7601	0.7903	0.8083	0.8109
	Ours	0.7951	0.8299	0.8376	<u>0.8394</u>
T → I	DCMH [11]	0.6277	0.6413	0.6491	0.6503
	SSAH [12]	0.6140	0.6285	0.6306	0.6324
	AGAH [7]	0.4240	0.5110	0.5329	0.5160
	JDSH [16]	0.6749	0.7155	0.7115	0.7181
	UKD [9]	0.6304	0.6562	0.6573	0.6639
	DCHUC ₂₀₂₂ [23]	0.7051	0.7243	0.7479	0.7453
	DAEH ₂₀₂₂ [19]	0.7130	0.7353	0.7455	0.7501
	SKDCH ₂₀₂₃ [20]	0.7228	0.7404	0.7519	0.7583
	CAGAN ₂₀₂₃ [15]	0.7350	0.7472	0.7676	0.7697
	UCMFH ₂₀₂₃ [25]	0.7939	0.8309	0.8306	0.8566
	CKDH ₂₀₂₄ [13]	0.7928	0.8354	0.8536	0.8435
	Ours	0.8258	0.8500	0.8562	0.8612

MAP Results: Table 1 and Table 2 show the Mean Average Precision (MAP) results of our method on the three widely used datasets. We compare two cross-modal retrieval tasks: $I \rightarrow T$ and $T \rightarrow I$: using image query texts and vice versa. Our proposed method outperforms the best deep unsupervised baseline CKDH [13] by 2.67% to 5.09% on MIRFlickr-25, and by 2.89% to 7.93% on IAPR TC-12, respectively. It achieves an improvement of up to 3.95% on NUS-WIDE. CKDH delivers prominent performances for both two retrieval tasks, but it achieves more performance improvements on the map of the $T \rightarrow I$ task than that of $I \rightarrow T$ task. Our method achieves almost the same results on both tasks, especially on the IAPR TC-12 dataset. The main reason may be that we use an Encoder with long text representation capability, which further validates the validity of our feature extraction method.

Table 3. The MAP@5000 performance of ablation experiments on MIRFlickr-25K and IAPR TC-12 datasets.

Task	Method	MIRFlickr-25K				IAPR TC-12			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	w/o Long	0.8625	0.8934	0.8992	0.9041	0.6360	0.6578	0.6768	0.6784
	w/o \mathcal{L}_q	0.8157	0.8627	0.8855	0.8917	0.6610	0.6917	0.6957	0.7079
	w/o \mathcal{L}_r	0.8489	0.8901	0.8919	0.9046	0.6581	0.6886	0.7088	0.7069
	Ours	0.8753	0.9092	0.9126	0.9151	0.6678	0.6922	0.7125	0.7097
$T \rightarrow I$	w/o Long	0.8697	0.8882	0.8957	0.9048	0.6378	0.6651	0.6867	0.6879
	w/o \mathcal{L}_q	0.8329	0.8653	0.8907	0.8993	0.6604	0.6890	0.6998	0.7074
	w/o \mathcal{L}_r	0.8521	0.8900	0.8903	0.8893	0.6594	0.6917	0.7093	0.7061
	Ours	0.8759	0.9107	0.9080	0.9134	0.6658	0.6933	0.7105	0.7096

Table 4. The MAP@5000 performance of ablation experiments on NUS-WIDE dataset.

Task	Method	NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits
$I \rightarrow T$	w/o Long	0.7707	0.7905	0.8130	0.8167
	w/o \mathcal{L}_q	0.7987	0.8186	0.8251	0.8305
	w/o \mathcal{L}_r	0.7643	0.8132	0.8166	0.8343
	Ours	0.7951	0.8299	0.8376	0.8394
$T \rightarrow I$	w/o Long	0.7740	0.7971	0.8189	0.8114
	w/o \mathcal{L}_q	0.8278	0.8464	0.8466	0.8508
	w/o \mathcal{L}_r	0.7876	0.8355	0.8340	0.8490
	Ours	0.8258	0.8500	0.8562	0.8612

Top-N Precision Curves: Figure 2 shows the top-N precision curves for the $I \rightarrow T$ and $T \rightarrow I$ tasks at 128 bits. In our experiments, on the MIRFlickr-25K

dataset, our method achieved a Precision of 99%, which is approximately 7% higher than the current state-of-the-art methods (CKDH). Similarly, on the NUS-WIDE dataset, our method demonstrated notable improvement, reaching a Precision of 90%, which is about 5% higher than existing methods, and consistently performed well across different N values. More importantly, as the N value increases, our method's performance remains stable on these datasets, further validating its broad applicability and significant potential for improvement.

4.3 Ablation Experiments

To validate the effectiveness of each module in our proposed method, we investigated the impact of each module on the model's performance. The specific results are shown in Table 3 and Table 4. We designed three variants: (1) **w/o Long**, which indicates that we used CLIP instead of the Long-CLIP; (2) **w/o \mathcal{L}_q** , which removed Long-CLIP-based Binary Code Fusion and Selection module and (3) **w/o \mathcal{L}_r** , which means we removed the Identity Semantic Preservation module. Based on the results in the tables, it can be observed that the performance significantly decreased after removing the respective modules, demonstrating the effectiveness of our proposed modules. By observing the result in Table 3 and Table 4, we can see that after removing Long-CLIP, performance declines significantly, especially in the IAPR TC-12 and NUS-WIDE datasets. The possible reason is that these two datasets contain more pairs of long text samples. And also we can see \mathcal{L}_q and \mathcal{L}_r contribute considerably to the performance especially on MIRflickr-25K.

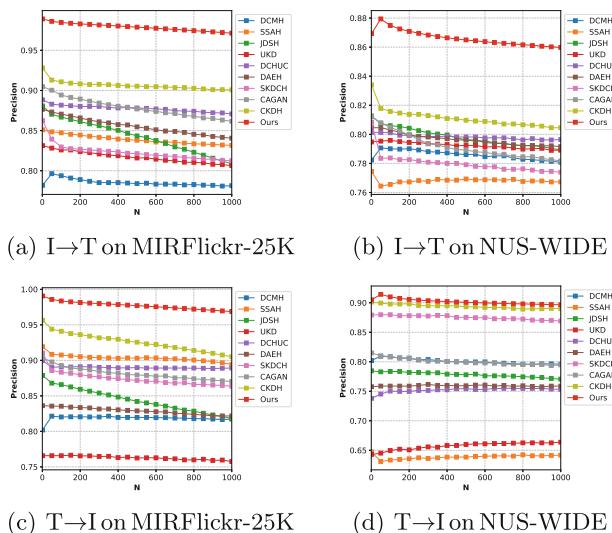


Fig. 2. Top-N precision curves@128 bits

4.4 Parameter Sensitivity Analysis

In our objective function, there are two hyperparameters, α and β , where α controls \mathcal{L}_r and β controls \mathcal{L}_q . To comprehensively understand their impact on the overall model, we conducted extensive experiments on three datasets. The results are illustrated in Fig. 3. For the MIRFlickr-25K and NUS-WIDE datasets, α ranges from 1 to 20 with an interval of 5, while β ranges from 1 to 5 with an interval of 1. For the IAPR TC-12 dataset, both α and β range from 0.1 to 0.5 with an interval of 0.1. The experimental results indicate that α has the most significant impact on the model's performance. We hypothesize that the primary reason for α 's prominent contribution is that the features extracted through the fine-tuned Encoder of Long-CLIP are prone to overfitting. By pre-extracting the features, we effectively mitigate this issue. Consequently, the adjustment of α plays a crucial role in balancing the model and avoiding overfitting.

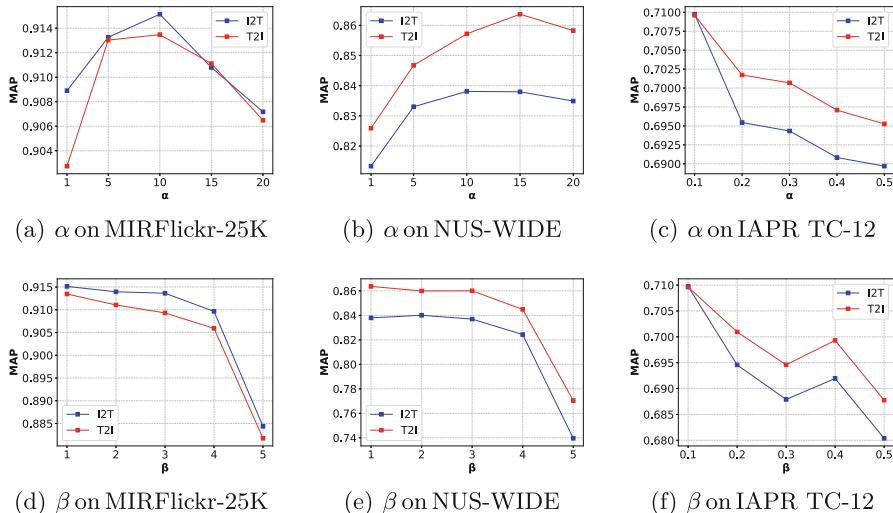


Fig. 3. Parameter sensitivity analysis@128 bits

5 Conclusion

In this paper, we introduced the Semantic Preservation and Hash Fusion Network (SPHFN) for cross-modal retrieval. Our method addresses the limitations of existing cross-modal retrieval techniques by effectively capturing long-range dependencies in text, preserving identity semantics, and constructing a joint similarity matrix for accurate and efficient retrieval. We conducted extensive experiments on three widely used benchmark datasets: MIRFlickr-25K, IAPR TC-12, and NUS-WIDE. The results demonstrated that SPHFN outperforms several state-of-the-art methods in terms of retrieval accuracy and precision.

The superior performance of SPHFN can be attributed to its innovative approach to semantic preservation and hash fusion, ensuring semantic consistency across modalities and enhancing retrieval effectiveness. Future work will focus on extending our approach to other cross-modal tasks and improving the scalability of the method to handle larger datasets and more complex retrieval scenarios. These advancements could further solidify the practical applications and robustness of SPHFN in various cross-modal retrieval tasks.

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

References

1. Chen, R., et al.: CLIP2Scene: towards label-efficient 3D scene understanding by clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7020–7030 (2023)
2. 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)
3. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, pp. 1–9 (2009)
4. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020)
5. Escalante, H.J., et al.: The segmented and annotated IAPR TC-12 benchmark. Comput. Vis. Image Underst. **114**(4), 419–428 (2010)
6. Ge, M., Li, Y., Wu, H., Li, M.: JM-CLIP: a joint modal similarity contrastive learning model for video-text retrieval. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3010–3014. IEEE (2024)
7. Gu, W., Gu, X., Gu, J., Li, B., Xiong, Z., Wang, W.: Adversary guided asymmetric hashing for cross-modal retrieval. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval, pp. 159–167 (2019)
8. 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)
9. Hu, H., Xie, L., Hong, R., Tian, Q.: Creating something from nothing: unsupervised knowledge distillation for cross-modal hashing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3123–3132 (2020)
10. Huiskes, M.J., Lew, M.S.: The MIR Flickr retrieval evaluation. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, pp. 39–43 (2008)
11. Jiang, Q.Y., Li, W.J.: Deep cross-modal hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3232–3240 (2017)
12. Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D.: Self-supervised adversarial hashing networks for cross-modal retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4242–4251 (2018)

13. Li, J., Wong, W.K., Jiang, L., Fang, X., Xie, S., Xu, Y.: CKDH: CLIP-based knowledge distillation hashing for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* (2024)
14. Li, M., Ma, L., Li, Y., Ge, M., et al.: CCAH: a clip-based cycle alignment hashing method for unsupervised vision-text retrieval. *Int. J. Intell. Syst.* **2023** (2023)
15. Li, Y., Ge, M., Li, M., Li, T., Xiang, S.: Clip-based adaptive graph attention network for large-scale unsupervised multi-modal hashing retrieval. *Sensors* **23**(7), 3439 (2023)
16. 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)
17. Lu, K., et al.: Deep unsupervised momentum contrastive hashing for cross-modal retrieval. In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 126–131. IEEE (2023)
18. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: *International Conference on Machine Learning*, pp. 8748–8763. PMLR (2021)
19. Shi, Y., et al.: Deep adaptively-enhanced hashing with discriminative similarity guidance for unsupervised cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **32**(10), 7255–7268 (2022)
20. Su, M., Gu, G., Ren, X., Fu, H., Zhao, Y.: Semi-supervised knowledge distillation for cross-modal hashing. *IEEE Trans. Multimedia* **25**, 662–675 (2021)
21. 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)
22. Tu, R.C., et al.: Unsupervised cross-modal hashing with modality-interaction. *IEEE Trans. Circuits Syst. Video Technol.* (2023)
23. Tu, R.C., et al.: Deep cross-modal hashing with hashing functions and unified hash codes jointly learning. *IEEE Trans. Knowl. Data Eng.* **34**(2), 560–572 (2020)
24. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
25. Xia, X., Dong, G., Li, F., Zhu, L., Ying, X.: When clip meets cross-modal hashing retrieval: a new strong baseline. *Inf. Fusion* **100**, 101968 (2023)
26. Xu, T., Liu, X., Huang, Z., Guo, D., Hong, R., Wang, M.: Early-learning regularized contrastive learning for cross-modal retrieval with noisy labels. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 629–637 (2022)
27. 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)
28. 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)
29. Yu, W., Liu, Y., Hua, W., Jiang, D., Ren, B., Bai, X.: Turning a clip model into a scene text detector. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6978–6988 (2023)
30. Zhang, B., Zhang, P., Dong, X., Zang, Y., Wang, J.: Long-CLIP: unlocking the long-text capability of clip. *arXiv preprint arXiv:2403.15378* (2024)
31. Zhang, J., Peng, Y., Yuan, M.: Unsupervised generative adversarial cross-modal hashing. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)

32. 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)
33. 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 Trans. Knowl. Data Eng.* (2022)

Machine Learning



Using High-Quality Feature for Weakly-Supervised Camouflaged Object Detection

Weijie Wu¹, Yiqiu Tong¹, Qijun Jiang², Lina Chen^{2(✉)}, and Hong Gao²

¹ College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China

² School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
chenlina@zjnu.cn

Abstract. Camouflaged objects appear almost identical to the background, which leads to low-level features carrying a large amount of noise, and redundant information being included in high-level features. Therefore, we propose the Detailed and Semantic Feature Extraction Network (DSNet) to recognize camouflaged objects from the background. Firstly, we use the Top-k Sparse Attention module (TKSA) to avoid interference from noise in low-level features. Secondly, we introduce the Spatial and Channel reconstruction Convolution module (SCConv) to remove redundant information in high-level features, at the same time, we propose the Asymmetric Logical Semantic Relation module (ALSR) to capture more semantic information in high-level features. Finally, extensive experiments show that our method achieves significant improvements compared to other weakly-supervised camouflaged object detection methods on three datasets.

Keywords: Weakly-Supervised · Camouflaged Object Detection · Feature Extraction

1 Introduction

Camouflaged objects have very similar visual properties to the background, which enables camouflaged objects to achieve perfect integration into the environment by deceiving the human visual system. Camouflaged object detection(COD) has become necessary research in many fields, such as biological discovery [9], medical image segmentation [8], industrial defect detection, and military camouflaged detection.

Research on camouflaged object detection has a long history, G.H. Thayer [23], H.B. Cott [19], and others' research on camouflaged animals has greatly promoted the development of camouflaged object detection. Early camouflaged object detection efforts were limited to handcrafted low-level features (color [1], texture [15]) to discriminate camouflaged objects from the background. With

the successful application of deep neural networks to the camouflaged object detection task, the performance of camouflaged object detection has been dramatically improved. Relevant researchers have also focused on the extraction of high-level features in addition to the extraction of low-level detailed features.

Mei et al. [14] proposed the Positioning and Focus Network (PFNet) consisting of a positioning module(PM) and a focus module(FM). PM extracts high-level semantic features by concatenating channel attention and spatial attention, and FM is used to remove false predictions caused by noise in low-level features. PFNet eliminates the effect of noise in low-level features but does not consider the effect of redundant information when extracting high-level semantic features. Dong et al. [4] proposed the practical receptive field module (DMC) to extract high-level semantic features from the perspective of a larger receptive field. DMC operates from the perspective of a larger receptive field to improve the semantic perception ability of the module but does not propose a relevant method to solve the noise problem in the low-level features.

Most camouflaged object detection methods cannot effectively solve the noise interference problem when extracting low-level detailed information. For the extraction of high-level semantic features, the impact of redundant information on the performance of camouflaged object detection is not considered, and the use of a fixed-size convolutional layer makes the network's semantic perception ability worse. Hence, we design a novel network to extract low and high-level critical features.

The main contributions are summarized as follows:

- A DSNet is proposed, which efficiently extracts low-level detailed features and high-level semantic features.
- The TKSA module is introduced to solve the problem of noise interference in low-level detailed feature extraction, which helps us focus on the most valuable detailed information.
- We introduce the SCConv module to address redundant features and propose an ALSR module for extracting semantic information. We concatenate two modules to extract more high-level semantic information.

2 Related Work

2.1 Detailed Feature Extraction

Detailed features help the network to accurately recognize the contours of camouflaged objects. Zhang et al. [29] fused the low-level features output from the encoding layer with the features obtained from the self-attention network to capture the critical detailed information, but the fusion process did not take into account the interference of noise. Chen et al. [2] proposed the Sparse Transformer to retain the most critical detailed features through a Top-k selection operator to adaptively retain the most critical detailed features; this method achieved better results in the field of image derain but was not applied in the field of camouflaged object detection. Xie et al. [20] used the detailed features

of infrared images fused with the detailed features of RGB images to obtain more expressive detailed features, but not all the datasets had infrared images. Wang [24] used a texture encoder to extract richer detailed features by adding the texture encoder to extract richer texture information but also increased the number of parameters in the network.

2.2 Semantic Feature Extraction

The key semantic features are extracted to help the network to localize the camouflaged region. Fan et al. [7] proposed a search module (SM) to extract high-level semantic information for localization. Since the Receptive Field Module (RF) in the search module is made up of convolutional layers of smaller size, it makes the network's semantic sensing capability weak. Sun et al. [22] proposed a Dual-branch Global Context Module(DGCM). DGCM extracts semantic information from the fused features across layers, but the features before fusion are obtained from the convolutional layer with a small sensory field, which makes the network semantically less perceptive. Pang et al. [17] proposed a mixed-scale triplet network (ZoomNet). ZoomNet mines semantic information at different scales through a scaling strategy, which allows the network to extract rich semantic information while introducing too much redundant information.

3 Detailed and Semantic Feature Extraction Network

The overall structure of DSNet is shown in Fig. 1. DSNet can not only extract low-level detailed features but also focus on high-level critical semantic information. The network contains the TKSA module, the SCConv module, and the ALSR module. The TKSA module extracts detailed features more accurately; The SCConv module helps the network to address redundant features; The ALSR module enables the network to capture more semantic information.

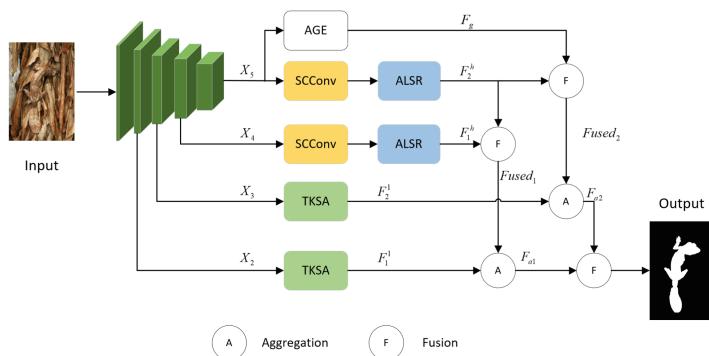


Fig. 1. The structure of DSNet

First of all, the image is fed into ResNet50 to get the output features $X_i (i = 1, 2, 3, 4, 5)$ in 5 stages. X_1 is the feature of the original image after only one down sampling block, which makes the extracted feature still very rough, so it also contains a lot of noise. If it is directly used for feature fusion, this will have a negative impact on the results. Since, only $X_i (i = 2, 3, 4, 5)$ are utilized in this thesis to further extract and aggregate critical features. The low-level output features X_2 and X_3 are fed into TKSA to extract low-level detailed information F_1^l and F_2^l . Meanwhile, the high-level output features X_4 and X_5 are processed by the SCConv module in tandem with the ALSR module to extract high-level semantic information F_1^h and F_2^h , respectively. Additionally, X_5 is passed through the Pyramid Pooling Module(AGE) [30] to obtain rich global information F_g . F_1^l and F_2^l contain abundant detailed information but less semantic information. The semantic information in F_1^h and F_2^h is rich, but there is less detailed information, so the perception ability of details is poor. To enable the network to extract features rich in detailed information and have abundant semantic information, we perform cross-level feature fusion and aggregation. Specifically, F_g and F_2^h are fused to produce global semantic information $F_{\text{Fused}2}$, which is then aggregated with F_2^l to obtain F_{a2} , which contains massive detailed information and significant semantic information; F_1^h and F_2^h are cross-level features fused to get $F_{\text{Fused}1}$, which contains semantic information of two layers, $F_{\text{Fused}1}$ and F_1^h are aggregated to get F_{a1} , which has rich detailed information and key semantic information; Finally, F_{a1} and F_{a2} are merged to obtain the Output, which contains all the level detailed information and semantic information.

3.1 Top-K Sparse Attention Module

The TKSA module [2] focuses on detailed information in camouflaged objects by capturing sparse features. The structure of the TKSA module is shown in Fig. 2.

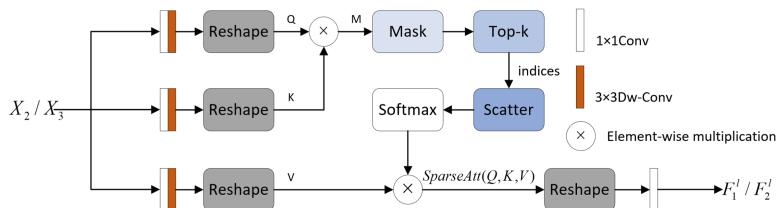


Fig. 2. The structure of TKSA

TKSA first adjusts the number of channels of $X_i (i = 2 \text{ or } 3)$ using 1×1 convolution layers. Compared to using convolution layers of other sizes to control the number of $X_i (i = 2 \text{ or } 3)$ channels, 1×1 convolution layers reduce the number of parameters. Subsequently, the height and width of the feature map are adjusted through 3×3 depth-wise convolutional layers, which have fewer parameters and

better computational efficiency compared to the standard 3×3 convolutional layers. Then, we perform a reshape operation on the output feature map to generate query (Q), key (K), and value (V). Next, it multiplies the transpose of Q and K to calculate the similarity matrix M between the query and key features of each pixel pair. Then, lower attention weights in the transposed attention matrix M are masked out. To avoid the interference of noise and irrelevant features, top-k scores are adaptively selected on the transposed attention matrix M so that the most important features are retained. Here, k is an adjustable parameter, which is determined by the weighted average. The scatter function sets the probability of elements less than top-k scores to 0 according to their indices, while keeping the other elements unchanged. Finally, the output is normalized by the softmax layer. The output value is given by $\text{softmax} \left(T_k \left(\frac{QK^\top}{\sqrt{d}} \right) \right)$. Multiplying $\text{softmax} \left(T_k \left(\frac{QK^\top}{\sqrt{d}} \right) \right)$ with V yields $\text{SparseAtt}(Q, K, V)$. The sparse attention representation is shown in Eq. (1):

$$\text{SparseAtt}(Q, K, V) = \text{softmax} \left(T_k \left(\frac{QK^\top}{\sqrt{d}} \right) \right) V \quad (1)$$

The learnable top-k selection operator $T_k(\cdot)$ is shown in Eq. (2), where Q , K , and V are in matrix form, K^T is the transpose matrix of K , and d is the dimension of Q and K . t_i is the k_{th} largest value in the j_{th} row of $\frac{QK^T}{\sqrt{d}}$.

$$[T_k(S)]_{ij} = \begin{cases} S_{ij} & S_{ij} \geq t_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Finally, we reshape the feature map of the sparse attention representation and perform a 1×1 convolution to adjust its dimension, ultimately obtaining the detailed features with noise removed.

3.2 Spatial and Channel Reconstruction Convolution Module

The SCConv module [12] solves the problem of feature redundancy in both spatial and channel dimensions. It consists of two parts: the Spatial Reconstruction Unit (SRU) for reducing spatial redundancy information and the Channel Reconstruction Unit (CRU) for reducing channel redundancy information. The structure of SCConv is illustrated in Fig. 3.

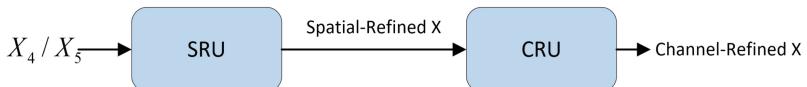


Fig. 3. The structure of SCConv

Spatial Reconstruction Unit Module. In order to remove feature redundancy in the spatial dimension, SRU separates redundant features in space through weight separation and then reconstructs the separated features using the cross-multiplication method to obtain spatially refined features. The specific structure of SRU is shown in Fig. 4.

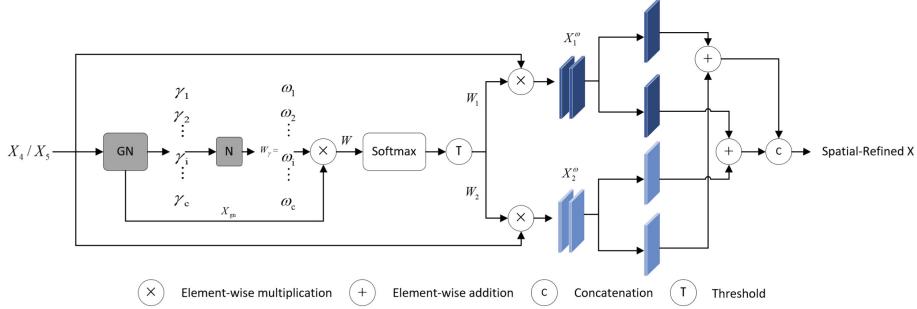


Fig. 4. The specific structure of SRU

The specific workflow of SRU can be divided into steps (a) to (g):

- (a) The input feature map $X_4/X_5 \in R^{B \times C \times H \times W}$ is Group normalized to obtain X_{gn} . The expression for X_{gn} is given by Eq. (3).

$$X_{gn} = GN(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (3)$$

μ and σ are the mean and standard deviation respectively in X_4/X_5 . ε is a constant, and here γ and β are trainable parameters.

- (b) The parameter $\gamma \in R^C$ (denoting the richness of spatial information on the feature map) in GN is normalized to obtain the importance mapping $W_\gamma \in R^C$ of different features, W_γ as shown in Eq. (4).

$$W_\gamma = \{\omega_i\} = \frac{\gamma_i}{\sum_{j=1}^C \gamma_j}, i, j = 1, 2, \dots, C \quad (4)$$

- (c) Multiply W_γ and X_{gn} to get the weighted feature map W , which represents the richness of spatial information on different feature maps, and normalize W using the softmax activation function.
- (d) By setting a threshold (0.5) and assigning weights greater than the threshold in the normalized W to 1, we obtain W_1 , which contains rich spatial information. Similarly, by resetting the weights that are 1 back to 0 (removing the spatial information-rich part), we obtain W_2 , which contains less spatial information.

- (e) By multiplying W_1 and W_2 with X_4/X_5 respectively, we obtain two weighted features. Feature X_1^ω , which indicates rich spatial information, and feature X_2^ω , which contains little spatial information.
- (f) X_2^ω contains less spatial information, but critical spatial information may still be present in it. To fully utilize the spatial information in X_2^ω , the cross-reconstruction method is employed to reconstruct both X_1^ω and X_2^ω .
- (g) The reconstructed X_1^ω and X_2^ω are concatenated to obtain spatial-refined feature X.

Channel Reconstruction Unit Module. CRU is primarily utilized to reduce the redundant information in the channel dimension. Firstly, the feature maps are split into two parts along the channel dimension, and then the feature maps of the two parts undergo feature extraction before being fused. The specific structure of CRU is shown in Fig. 5.

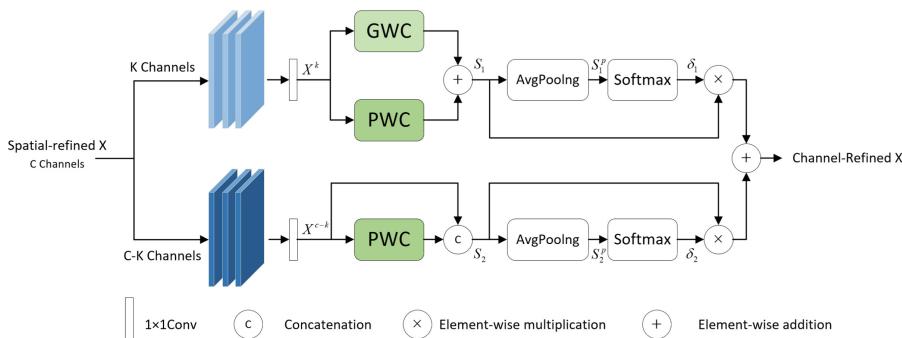


Fig. 5. The structure of CRU

CRU initially divides the C-channel spatial-refined feature X into a K-channel feature map and a C-K-channel feature map in terms of channel dimension, with K taking the value range of $[0, C]$. Subsequently, a 1×1 convolutional layer controls the number of channels in the two parts of the feature maps, resulting in X^k and X^{c-k} .

X^k is separately input into GWC and PWC for convolutional operations. GWC reduces parameters while extracting more representative features from X^k . PWC enhances inter-channel information linkage within X^k . The transformed mappings from PWC and GWC are then summed, yielding the more representative feature S_1 at a smaller computational cost.

Simultaneously, X^{c-k} is input into PWC to capture detailed features. To avoid missing features, X^{c-k} is concatenated with the features processed by PWC, resulting in the detailed feature S_2 .

Next, global average pooling is applied to both features S_1 and S_2 , resulting in spatial information S_1^P and S_2^P for different channels. These are then fed into

the softmax activation function to generate important feature vectors δ_1 and δ_2 for the channel ranges, respectively.

Multiplying δ_1 and δ_2 with S_1^P and S_2^P yields refined features for the K-channel and C-K-channel ranges. Finally, the channel-refined features are summed along the channel direction to obtain Channel-Refined X.

3.3 Asymmetric Logical Semantic Relation Module

ALSR can capture more semantic information when extracting high-level features. It consists of multiple feature extraction branches, each containing different sizes and quantities of asymmetric convolutional blocks (ACBlocks) [3]. The method of parallel connection between asymmetric convolution and standard convolution in ACblock allows the module to obtain a larger receptive field without increasing too many parameters, thereby capturing long-distance semantic information effectively. The structure of the ALSR module is shown in Fig. 6.

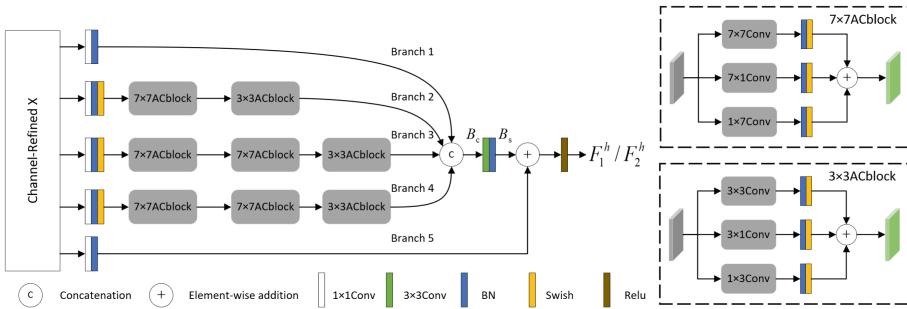


Fig. 6. The structure of ALSR

ALSR initially feeds the input features Channel-Refined X into Branch i ($i = 1, 2, 3, 4, 5$) for extracting semantic information at different scales, respectively. Branch 1 and Branch 5 are composed of a 1×1 convolutional layer for adjusting the number of channels and a Batch Normalization layer. Branch 1 and Branch 5 mainly serve to prevent the loss of information due to excessive layers of convolution during the feature extraction. Branch 2, 3, and 4 adjust the number of channels by 1×1 convolution of the input features. Subsequently, the adjusted feature maps are normalized by the BN layer and then undergo nonlinear transformation via the Swish activation function. Finally, asymmetric convolution blocks of different sizes and numbers are used to extract the semantic information from the feature maps.

We concatenate Branch i ($i = 1, 2, 3, 4$) along the channel dimension to obtain B_c , which contains a large amount of semantic information. Next, B_c is passed through a 3×3 convolutional layer to extract critical semantic information and

reduce the number of channels, followed by a Batch Normalization layer to obtain B_s . To mitigate potential information loss resulting from excessive convolutional layers, we combine the normalized feature B_s with the output feature of Branch 5 and apply the ReLU activation to obtain high-level semantic information F_1^h/F_2^h .

The specific structure of the ACblock is shown in Fig. 6. A 7×7 ACblock is formed by the following three branches in parallel, respectively: (1) A branch consisting of a 7×7 convolution layer, a BN layer, and a Swish activation function connected in series. (2) A branch consisting of a 7×1 convolution layer, a BN layer, and a Swish activation function connected in series. (3) A branch consisting of a 1×7 convolution layer, a BN layer, and a Swish activation function in series. The structure of the 3×3 ACblock is similar.

Each ACblock consists of three branches. Compared to using only one standard convolution branch, ACblocks can focus on larger receptive fields. Since asymmetric convolution is used, the number of parameters will not significantly increase. It is precisely for this reason that the ALSR module can help the network obtain a larger receptive field, thereby further improving the network's semantic perception ability.

4 Experiments

4.1 Datasets and Evaluation Criteria

We train our network on the SCOD dataset [10], which comprises 4040 weakly supervised labeled images (3040 images from the COD-10K dataset and 1000 images from the CAMO dataset). The trained network is then tested on three public datasets (CHAMELEON dataset [21], CAMO dataset [11], and COD-10K dataset [7]).

Four widely used evaluation metrics are employed to evaluate the performance of camouflaged object detection: Mean Absolute Error (MAE) [18], which evaluates the pixel-level accuracy between the predicted image and the ground truth; mean E-measure (E_m) [6], the E_m metric is used to evaluate the accuracy of the camouflaged object detection results, both locally and as a whole; S-measure (S_m) [5], which serves as an evaluation metric for the structural similarity of camouflaged object detection; Weighted F-measure(F_β^ω) [13], the F_β^ω metric is the weighted average of precision and recall, which is used to comprehensively evaluate the overall performance of camouflaged object detection by balancing precision and recall.

4.2 Implementation Details

The proposed DSNet is implemented in PyTorch. Firstly, the input image is resized to 320×320 and trained using the SGD optimizer with a momentum of 0.9, a weight decay of $5e-4$, and a maximum learning rate of $1e-3$. The training phase is initialized with the seed set to 3407 and a batch size of 16, running for a

total of 150 epochs, which takes about 3 h. During the inference stage, the image is simply resized to 320×320 before making the prediction, with an inference speed of approximately 80 FPS. The experiments are conducted on the following platform: Intel® i9-12900K CPU @ 3.90 GHz \times 16, GeForce RTX 3090 GPU.

4.3 Comparison Experiments

To demonstrate the effectiveness of our method, we compare it with four fully supervised camouflaged object detection methods, two unsupervised camouflaged object detection methods, and three weakly supervised camouflaged object detection methods.

Table 1 shows that our method outperforms weakly supervised and unsupervised camouflaged object detection methods on three datasets. Compared to the best-performing SCOD method [10] among weakly supervised methods, our method achieves an improvement of 0.4%, 0.2%, and 0.6% in the S_m metric on the CHAMELEON dataset(76 images), CAMO dataset(250 images), and COD-10K dataset(2026 images), respectively; the F_β^ω metric obtains 0.4%, 0.7%, and 1.1% improvement, respectively; the MAE and E_m metrics maintain their optimal values and slightly improved. Compared with unsupervised methods, the performance of this method has been significantly improved, indicating that it exhibited better performance in detecting camouflaged objects. Although there is still a certain gap compared to the fully supervised method, having a supervised method is bound to incur more time costs.

To more intuitively demonstrate the superior performance of our method compared to the current best weakly supervised camouflaged object detection methods, we plotted PR(Precision-Recall) curves and F-measure curves. The PR curves and F-measure curves are shown in Fig. 7 and Fig. 8.

Table 1. In addition to the fully supervised method, the best-performing values on each dataset are bolded in bold, and the second-best-performing values are labeled with “-”. “F”, “U” and “W” denote fully supervised, unsupervised, and weakly supervised, respectively.

Methods	Sup	CHAMELEON				CAMO				COD-10K			
		MAE↓	Sm↑	Em↑	$F_\beta^\omega \uparrow$	MAE↓	Sm↑	Em↑	$F_\beta^\omega \uparrow$	MAE↓	Sm↑	Em↑	$F_\beta^\omega \uparrow$
SINet	F	0.044	0.869	0.899	0.740	0.100	0.751	0.834	0.606	0.051	0.771	0.797	0.551
PFNet	F	0.033	0.882	0.942	0.810	0.085	0.782	0.852	0.695	0.040	0.800	0.868	0.660
UGTR [25]	F	0.031	0.888	0.910	0.794	0.086	0.784	0.822	0.684	0.036	0.817	0.852	0.666
ZoomNet	F	0.023	0.902	0.958	0.845	0.066	0.820	0.892	0.752	0.029	0.838	0.911	0.729
DUSD [28]	U	0.129	0.578	0.632	0.312	0.166	0.556	0.594	0.308	0.107	0.580	0.646	0.276
USPS [16]	U	0.207	0.568	0.641	0.398	0.188	0.573	0.631	0.380	0.196	0.519	0.536	0.266
SS [27]	W	0.067	0.782	0.860	0.654	0.118	0.696	0.786	0.562	0.071	0.684	0.770	0.461
LSC [26]	W	0.053	0.792	0.881	0.714	0.102	0.713	0.795	0.618	0.055	0.710	0.805	0.546
CRNet [10]	W	<u>0.046</u>	<u>0.818</u>	<u>0.897</u>	<u>0.744</u>	0.092	<u>0.735</u>	0.815	<u>0.641</u>	0.049	<u>0.733</u>	0.832	<u>0.576</u>
Ours	W	0.044	0.822	0.910	0.748	0.092	0.737	0.815	0.648	0.049	0.739	0.832	0.587

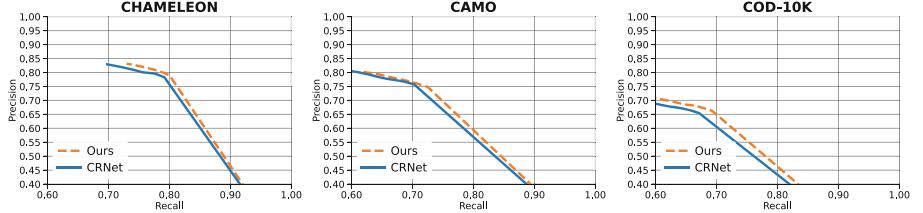


Fig. 7. PR curves

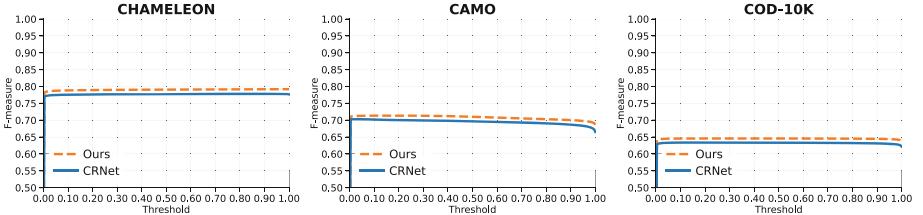


Fig. 8. F-measure curves

4.4 Ablation Studies

To verify the effectiveness of each module, we conduct ablation experiments on CHAMELEON dataset (76 images), CAMO dataset (250 images) and COD-10K dataset (2026 images). The results of the ablation experiments are shown in Table 2.

Table 2. We labeled the best-performing value on each dataset in bold and the second best-performing value in “-”. “M”, “B”, “A”, “S” and “T” denote “Methods”, “Basic Network”, “ALSR”, “SCConv” and “TKSA” respectively.

M	B	A	S	T	CHAMELEON				CAMO				COD-10K			
					MAE↓	Sm↑	Em↑	$F_{\beta}^{\omega} \uparrow$	MAE↓	Sm↑	Em↑	$F_{\beta}^{\omega} \uparrow$	MAE↓	Sm↑	Em↑	$F_{\beta}^{\omega} \uparrow$
1	✓				0.050	0.804	0.883	0.727	0.103	0.722	0.807	0.621	0.052	0.727	0.822	0.566
2	✓	✓			<u>0.045</u>	0.819	<u>0.902</u>	0.744	<u>0.096</u>	0.728	0.809	0.634	0.049	0.730	0.822	0.573
3	✓		✓		0.047	0.809	0.892	0.734	0.101	0.722	0.809	<u>0.641</u>	<u>0.050</u>	0.729	0.823	0.571
4	✓			✓	0.046	0.811	0.892	0.732	0.100	0.722	0.808	0.622	0.051	0.729	0.823	0.568
5	✓	✓	✓		0.046	0.814	0.901	0.737	<u>0.096</u>	<u>0.729</u>	0.807	0.636	<u>0.050</u>	<u>0.737</u>	0.832	<u>0.584</u>
6	✓	✓		✓	<u>0.045</u>	<u>0.82</u>	0.910	<u>0.745</u>	0.097	0.726	<u>0.810</u>	0.634	<u>0.050</u>	0.734	<u>0.828</u>	0.577
7	✓		✓	✓	0.046	0.811	0.901	0.733	0.100	0.726	0.805	0.632	<u>0.050</u>	0.730	<u>0.828</u>	0.572
Ours	✓	✓	✓	✓	0.044	0.822	0.910	0.748	0.092	0.737	0.815	0.648	0.049	0.739	0.832	0.587

We conduct ablation experiments on three datasets, and when all the modules we introduced and designed are put into the network at the same time, the performance of our network is the best.

Since the test images of COD-10K are far more than two other datasets, we only analyze the evaluation indicators on COD-10K. It can be clearly seen from Table 2 that Method 2, 3, and 4 have improved in all four evaluation indicators compared to Method 1, proving that ALSR, SCConv and TKSA are both beneficial for improving the performance of camouflaged object detection.

Method 5 connects two modules, SCConv and ALSR, in tandem for extracting high-level semantic information. Method 5 compares with Method 2, which is the best performer among Method 2, 3, and 4, which only incorporates one module, and the MAE metric of Method 5 remains almost unchanged while the S_m metric is improved by 0.7%, the E_m metric is improved by 1%, and the F_β^ω metric is improved by 1.1%.

Method 6 and Method 7 only used ALSR or SCConv to extract high-level semantic information, while TKSA is used to extract low-level detailed information. The best performing Method 6 among Method 6 and Method 7 is also inferior to Method 5. Compared with Method 6, the MAE metric of Method 5 remains unchanged, the S_m metric has increased by 0.3%, the E_m metric has increased by 0.4%, and the F_β^ω metric has increased by 0.7%. This also proves that the use of SCConv in tandem with ALSR helps extract high-level semantic information and improve the performance of camouflaged object detection.

In our approach, the tandem of SCConv and ALSR is used to extract high-level semantic information and TKSA is used to extract low-level detailed information. The experimental results of ours are optimal in terms of the values of the four evaluation metrics, which further verified that the simultaneous use of the three modules maximized the performance of camouflaged object detection.

5 Conclusion

Considering the feature extraction of the images by the backbone network, the feature maps outputted by different levels contain different features. The low-level feature contains abundant noise, while the high-level feature contains redundant information. Firstly, we introduce the TKSA module to filter out the noise during the extraction of low-level features, so as to capture more accurate and detailed features. Secondly, we employ the SCConv module and ALSR module to extract refined semantic information from the high-level features. Finally, comparison experiments on three public datasets demonstrate that our proposed method outperforms many weakly-supervised camouflaged object detection methods. Method experiments are conducted on COD-10K, and the experimental results prove that our introduced and proposed modules are effective in improving the performance of camouflaged object detection.

At present, most camouflaged object detection researchers focus on improving detection accuracy, recall rate and other evaluation indicators. However, when applying the camouflaged object detection to a series of downstream tasks, such as agricultural pest detection, industrial defect detection, etc. We should also fully consider the detection speed and hardware requirements.

Acknowledgement. This study was supported by the Key Project of Regional Innovation and Development Joint Fund of National Natural Science Foundation of China (Grant No. U22A2025).

References

1. Boot, W.R., Neider, M.B., Kramer, A.F.: Training and transfer of training in the search for camouflaged targets. *Attention Percept. Psychophys.* **71**, 950–963 (2009)
2. Chen, X., Li, H., Li, M., Pan, J.: Learning a sparse transformer network for effective image DeRaining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5896–5905 (2023)
3. Ding, X., Guo, Y., Ding, G., Han, J.: ACNet: strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1911–1920 (2019)
4. Dong, B., Zhuge, M., Wang, Y., Bi, H., Chen, G.: Accurate camouflaged object detection via mixture convolution and interactive fusion. arXiv preprint [arXiv:2101.05687](https://arxiv.org/abs/2101.05687) (2021)
5. Fan, D.P., Cheng, M.M., Liu, Y., Li, T., Borji, A.: Structure-measure: a new way to evaluate foreground maps. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4548–4557 (2017)
6. Fan, D.P., Gong, C., Cao, Y., Ren, B., Cheng, M.M., Borji, A.: Enhanced-alignment measure for binary foreground map evaluation. arXiv preprint [arXiv:1805.10421](https://arxiv.org/abs/1805.10421) (2018)
7. Fan, D.P., Ji, G.P., Sun, G., Cheng, M.M., Shen, J., Shao, L.: Camouflaged object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2777–2787 (2020)
8. Fan, D.-P., et al.: PraNet: parallel reverse attention network for polyp segmentation. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12266, pp. 263–273. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59725-2_26
9. Pérez-de la Fuente, R., et al.: Early evolution and ecology of camouflage in insects. *Proc. Nat. Acad. Sci.* **109**(52), 21414–21419 (2012)
10. He, R., Dong, Q., Lin, J., Lau, R.W.: Weakly-supervised camouflaged object detection with scribble annotations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 781–789 (2023)
11. Le, T.N., Nguyen, T.V., Nie, Z., Tran, M.T., Sugimoto, A.: Anabranch network for camouflaged object segmentation. *Comput. Vis. Image Underst.* **184**, 45–56 (2019)
12. Li, J., Wen, Y., He, L.: SCConv: spatial and channel reconstruction convolution for feature redundancy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6153–6162 (2023)
13. Margolin, R., Zelnik-Manor, L., Tal, A.: How to evaluate foreground maps? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2014)
14. Mei, H., Ji, G.P., Wei, Z., Yang, X., Wei, X., Fan, D.P.: Camouflaged object segmentation with distraction mining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8772–8781 (2021)
15. Neider, M.B., Zelinsky, G.J.: Searching for camouflaged targets: effects of target-background similarity on visual search. *Vision. Res.* **46**(14), 2217–2235 (2006)

16. Nguyen, T., et al.: DeepUSPS: deep robust unsupervised saliency prediction via self-supervision. In: *Advances in Neural Information Processing Systems*, vol. 32 (2019)
17. Pang, Y., Zhao, X., Xiang, T.Z., Zhang, L., Lu, H.: Zoom in and out: a mixed-scale triplet network for camouflaged object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2160–2170 (2022)
18. Perazzi, F., Krähenbühl, P., Pritch, Y., Hornung, A.: Saliency filters: contrast based filtering for salient region detection. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–740. IEEE (2012)
19. Poulton, E.: *Adaptive Coloration in Animals* (1940)
20. Shuchun Xie, Z.C., Sheng, B.: RGB-IR multi-channel feature fusion semantic segmentation network with enhanced details. *Comput. Eng.* **48**(10), 230–237+244 (2022)
21. Skurowski, P., Abdulameer, H., Błaszczyk, J., Depta, T., Kornacki, A., Koziel, P.: Animal camouflage analysis: Chameleon database. *Unpublished manuscript* **2**(6), 7 (2018)
22. Sun, Y., Chen, G., Zhou, T., Zhang, Y., Liu, N.: Context-aware cross-level fusion network for camouflaged object detection. *arXiv preprint arXiv:2105.12555* (2021)
23. Thayer, G.H.: *Concealing-Coloration in the Animal Kingdom: An Exposition of the Laws of Disguise Through Color and Pattern*. Macmillan Company (1918)
24. Wang, X.: Research on camouflage target detection and military camouflage detection algorithms based on deep learning. *Master's thesis*, Jilin University (2024)
25. Yang, F., et al.: Uncertainty-guided transformer reasoning for camouflaged object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4146–4155 (2021)
26. Yu, S., Zhang, B., Xiao, J., Lim, E.G.: Structure-consistent weakly supervised salient object detection with local saliency coherence. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 3234–3242 (2021)
27. Zhang, J., Yu, X., Li, A., Song, P., Liu, B., Dai, Y.: Weakly-supervised salient object detection via scribble annotations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12546–12555 (2020)
28. Zhang, J., Zhang, T., Dai, Y., Harandi, M., Hartley, R.: Deep unsupervised saliency detection: a multiple noisy labeling perspective. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9029–9038 (2018)
29. Zhang, M., Xu, S., Piao, Y., Shi, D., Lin, S., Lu, H.: PreyNet: preying on camouflaged objects. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5323–5332 (2022)
30. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890 (2017)



ECHO: Adaptive Correction for Subgraph-Wise Sampling with Lightweight Hyperparameter Search

Dingwei Liu, Zhenyu Li^(✉), and Zhibin Zhang

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
`{liudingwei19g, zyli, zhangzhibin}@ict.ac.cn`

Abstract. Graph neural networks (GNNs) excel in a wide range of complex graph-based tasks. When training GNNs on large datasets, subgraph-wise sampling (SS) methods have shown remarkable speed-up per epoch by avoiding recursive sampling in node-wise sampling (NS) ones. However, this comes at the cost of a larger variance in estimating stochastic gradients, resulting in either slow convergence or low accuracy. In this paper, we first delve into the variance of SS methods and reveal that it can be further decoupled into two components: *graph drop variance* and *skewed sample variance*. We then propose ECHO that mitigates the negative impact of variances, in order to achieve comparable accuracy to node-wise sampling while greatly accelerating training in large datasets. Specifically, ECHO introduces a variance-aware hyperparameter search algorithm that leverages lightweight variance estimation in the preprocessing stage. In the training stage, ECHO adaptively incorporates neighbor sampling epochs into the training process as correction according to the training loss. Our theoretical analysis and extensive experiments demonstrate that ECHO achieves fast convergence with high accuracy. Specifically, ECHO achieves up to $3.5\times$ training time speed-up and comparable accuracy compared to node-wise sampling; in comparison with SS baselines, ECHO achieves the best accuracy and up to $11\times$ convergence speed-up.

Keywords: Graph neural network · Subgraph-wise sampling · Correction

1 Introduction

Graph Neural Networks (GNNs) exhibit robust performance on graph datasets covering various tasks, *e.g.*, social recommendation [10, 15], knowledge graph processing [9, 16], drug discovery [18, 20]. However, training GNNs on large real-world graphs, like the Facebook social network [1] with 3 billion nodes, is challenging on a single GPU. Since GNNs recursively aggregate features according to the graph structure, the straightforward mini-batch training still involves a substantial portion of the graph, causing significant resource consumption in node feature slicing and data transmission from CPU to GPU.

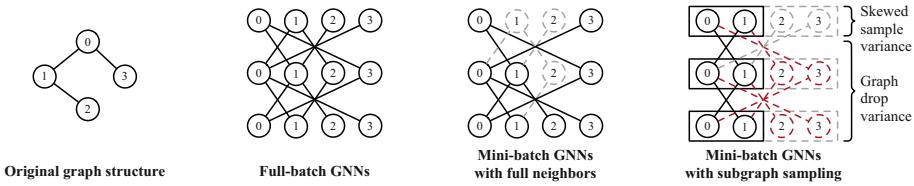


Fig. 1. Comparing batch subgraph examples of full-batch versus mini-batch with full neighbors and subgraph-wise sampling (SS). Black denotes nodes/edges in training subgraph; gray denotes nodes/edges not involved in training subgraph; red denotes the adjacent nodes/edges that SS drops in training subgraph (Color figure online)

To mitigate this problem, various sampling-based methods including *Node-wise sampling (NS)* [10], *Layer-wise sampling (LS)* [2, 23], and *Subgraph-wise sampling (SS)* [3, 8, 21] have been proposed. Among them, *NS* achieves the best convergence and accuracy but with a higher time cost due to sampling and data transmission. *LS* may suffer significant accuracy loss due to the sparsity created in training subgraphs [6]. By avoiding recursive neighbor expansion during sampling, *SS* reduces the size of training subgraphs and improves training efficiency per epoch. As a price, *SS* shows a decline in accuracy and convergence compared to *NS*.

To further explore the potential of *SS*-based methods, we analyze the variance introduced by *SS* that hurts performance. We decouples the *SS* variance in estimating stochastic gradients into two components: *graph drop variance* and *skewed sample variance*. *Graph drop variance* is due to training subgraphs generated by *SS* that ignore out-of-partition nodes and edges. *Skewed sample variance* is due to skewed sampling of training nodes. As shown in Fig. 1, training nodes in the same partition can only be sampled together by *SS*.

Although existing works deploy varying design choices (e.g., Metis [12] in preprocessing) to mitigate the negative impact, *SS* variance inevitably affects convergence and accuracy. One can make up for such variance by incorporating correction epochs [5, 17] that introduce additional information. However, a challenge that has not been touched when using correction in *SS* training is when the correction epochs should be executed: inadequate correction epochs neither accelerate convergence nor improve accuracy, while excessive correction epochs prolong overall training time due to the overhead introduced by correction. Simply using periodic correction with fixed or increasing frequency cannot balance training time and accuracy.

To address the above gap, we propose ECHO that consists of lightweight hyperparameter search and adaptive correction for *SS*-based GNN training. In the preprocessing stage, ECHO introduces a variance-aware hyperparameter search (VHS) algorithm. VHS uses the simulation of actual batch node sets to capture the level of *graph drop variance* and *skewed sample variance*. The simulation is lightweight thanks to *continuous partitioning* [22] that enables logical node set partitioning (as opposed to real partitioning). In the training stage,

ECHO introduces an adaptive epoch switching (AES) algorithm that incorporates correction into the SS-based training process to mitigate the impact of SS variance. AES adaptively guide training switching between SS epochs and correction based on approximate training loss. Specially, ECHO utilizes mini-batch epochs with neighbor sampling as the correction. Our theoretical analysis shows that, with certain correction epochs, ECHO can achieve a convergence rate of $\mathcal{O}(1/\sqrt{T})$, where T is the total epoch number.

We implement ECHO in PyG [7] and compare it with state-of-the-art sampling based methods on real-world graph datasets. The ablation experiments further demonstrate the effectiveness of each component in ECHO. In summary, the main contributions of this work are as follows:

- We theoretically analyze the convergence of subgraph-wise sampling based methods. The analysis reveals the two components of variance that affect training performance.
- We propose ECHO, a subgraph-wise sampling based training method for fast training convergence and improved accuracy. ECHO consists of a variance-aware hyperparameter search for preprocessing and an adaptive correction algorithm for training. We also provide theoretical analysis of ECHO’s convergence.
- Extensive evaluations show that ECHO achieves up to $3.5\times$ speed-up in terms of training time and comparable accuracy compared to node-wise sampling. In comparison to SS baselines, ECHO achieves the best accuracy and up to $11\times$ convergence speed-up.

The code and supplementary materials are publicly available at <https://github.com/loc-l/ECHO>.

2 Background and Related Works

Vanilla Training for GNNs. Kipf *et al.* [13] first propose full-batch training for GNN, which minimizes the empirical loss over all training node. To deal with large graphs, Hamilton *et al.* [10] use the mini-batch training method, which is heavily used for non-graph neural network training (*e.g.*, for CNNs). Intuitively, each GNN mini-batch i.i.d samples nodes from the training set, and computes with all their neighbors.

Sampling-Based Training for GNNs. Various sampling methods that aim at reducing batch training subgraph size have been proposed [2, 3, 8, 10, 21, 23]. *Node-wise sampling (NS)* method neighbor sampling [10] fixes neighborhood size per layer. It converges fast with good prediction performance, but at a high cost per epoch due to iterative sampling and data transmission. *Layer-wise sampling (LS)* methods [2, 23] sample constant number of nodes for each layer. FastGCN [2] is time-efficient at the cost of sparse batch adjacency matrices leading to poor accuracy. LADIES [23] develops a top-down layer-dependent node sampling that maintains denser adjacency matrices at the cost of significantly

more computation for sampling. *Subgraph-wise sampling (SS)* methods execute full-batch training on induced batch subgraphs. Cluster-GCN [3] uses clustering-based algorithms (*e.g.*, Metis [12]) to partition the graph as preprocessing. For training a batch, it randomly samples multiple partitions and merges them into a training subgraph. Cluster-GCN avoids recursive neighbor access, allowing control over the number of transmitted nodes per epoch to the total graph nodes. To access more structural information, GAS [8] includes full 1-hop neighbors of the training nodes in each batch and uses historical embeddings of out-partition neighbors to avoid extra computation. GraphSAINT [21] uses properly designed samplers (*e.g.*, random walk sampler) to sample node sets and induce them to subgraphs (partitions), one of which will be sampled as a training subgraph per batch.

Summary. SS significantly reduces the time per epoch, because it breaks the dependency between nodes across partitions. As such, SS-based methods improve computational efficiency and reduce the time of transmission between CPU and GPU. Nevertheless, this comes at the cost of noticeable accuracy drop and convergence slow-down compared to node-wise sampling method. The increase in training epochs makes the overall training speed-up weakened. To further explore the potential of SS, we will analyze the main factors that affect its performance in the next section.

3 Analysis for Subgraph-Wise Sampling

3.1 Residual Error of Subgraph-Wise Sampling

Inspired by the decomposition of the mean-square error (MSE) of stochastic gradient in [4], we analyze the variances that subgraph-wise sampling introduces by decomposing the MSE of the stochastic gradient of subgraph-wise sampling (SS) training and vanilla mini-batch training.

Let $\nabla \mathcal{L}_B(\theta)$ and $\nabla \mathcal{L}_P(\theta)$ to denote the stochastic gradient of vanilla mini-batch training and subgraph-wise sampling training, respectively. To facilitate analysis, we introduce stochastic gradient $\nabla \mathcal{L}_P^{\text{full}}(\theta)$, which uses training nodes generated by subgraph-wise sampling and computes embeddings with full neighborhoods in the original graph. Then, the decomposition of MSE of the stochastic gradient is as follows:

$$\begin{aligned} & \frac{1}{2} \mathbb{E} \left[\|\nabla \mathcal{L}_P(\theta_t) - \nabla \mathcal{L}_B(\theta_t)\|^2 \right] \\ & \leq \underbrace{\mathbb{E} \left[\|\nabla \mathcal{L}_P(\theta_t) - \nabla \mathcal{L}_P^{\text{full}}(\theta_t)\|^2 \right]}_{\kappa_1^2(t)} + \underbrace{\mathbb{E} \left[\|\nabla \mathcal{L}_P^{\text{full}}(\theta_t) - \nabla \mathcal{L}_B(\theta_t)\|^2 \right]}_{\kappa_2^2(t)} \end{aligned} \quad (1)$$

Let $\kappa_1^2 = \max_t \kappa_1^2(t)$, $\kappa_2^2 = \max_t \kappa_2^2(t)$ to be the upper bound of each, respectively. We call κ_1^2 the *graph drop variance*, as it is caused by dropping out-partition graph structure including nodes and edges. We call κ_2^2 the *skewed sample variance*, as it is caused by skewed sampling of the training set. Subgraph-wise

sampling selects all training nodes within the same partition or selects none of them.

We next analyze how graph drop variance and skewed sample variance affect the convergence of mini-batch training with subgraph-wise sampling. Before that, we make the following standard assumptions as [4, 6, 17].

Assumption 1. *The loss function $\mathcal{L}(\cdot, \cdot)$ is $C_{\mathcal{L}}$ -Lipschitz continuous and $L_{\mathcal{L}}$ -smoothness w.r.t. to the input node embedding vector.*

Assumption 2. *The activation function $\sigma(\cdot, \cdot)$ is C_{σ} -Lipschitz continuous and L_{σ} -smoothness.*

Following the definition of κ_1^2, κ_2^2 , we analyze the expectation of $\nabla \mathcal{L}(\theta_t)$ after T epochs.

Theorem 1 (*Convergence of subgraph-wise sampling*). *Suppose Assumption 1 and 2 hold. If we choose the learning rate $\eta = \frac{1}{\sqrt{T}}$, then we have:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla \mathcal{L}(\theta_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}(\kappa_1^2 + 2\kappa_2^2) \quad (2)$$

Proof. The details on the proof is provided in supplementary materials.

Theorem 1 implies that the gradient of SS-based GNN training is bounded by $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}(\kappa_1^2 + 2\kappa_2^2)$. The irreducible residual error $\mathcal{O}(\kappa_1^2 + 2\kappa_2^2)$ hurts the convergence. That said, the convergence of SS-based training suffers from graph drop variance and skewed sample variance, regardless of the generation strategies of partitions.

3.2 Motivations and Challenges

In existing works, Metis [12] (used in Cluster-GCN [3] and GAS [8]) and random walk sampler (used in GraphSAINT [21]) can mitigate graph drop variance by generating more densely connected subgraphs for training, while Cluster-GCN [3] and GAS [8] select multiple parts for each batch to mitigate skewed sample variance. However, the efficacy of these subgraph generation strategies depends on the level of introduced SS variance across different datasets. As shown in our experiments, these SS-based methods [3, 8, 21] suffer from varying levels of convergence slow-down and accuracy drop. Unfortunately, existing works lack metrics to evaluate the efficiency of training subgraph generation strategy without actual training (**C1**). We thus turn to establishing lightweight variance estimation to specify subgraph generation strategy.

As explained in Sect. 3.1, the residual error hurts performance at varying levels, even with reduced SS variance. We come to mitigate its impact in the training stage using the idea of correction by periodically refining the learned

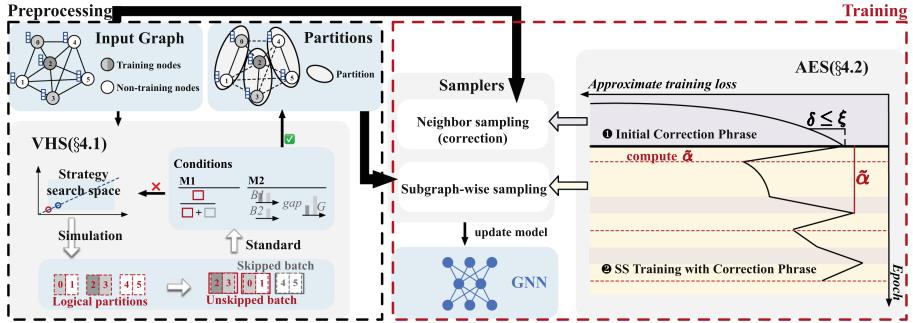


Fig. 2. Overview of ECHO. In the preprocessing stage, ECHO conducts variance-aware hyperparameter search (VHS), which simulates the training batch nodes to check conditions (**M1**, **M2**) for corresponding variance. In the training stage, ECHO leverages adaptive epoch switch (AES) to guide training, where the purple and yellow demonstrate correction and SS, respectively. The red dotted lines depicts the computation of the conditions for incorporating one epoch of correction (Color figure online)

model [5, 17]. In the context of SS, the introduction of correction can improve convergence and accuracy by breaking the structural information restriction of partitions and reducing the impact of skewed sampling. However, existing correction strategies show various performance on datasets with distinct degrees of SS variance. This is because inappropriate correction cannot effectively accelerate convergence, and may prolong training time due to the increased overhead required for correction. Simply using periodical correction across different datasets cannot yield a trade-off between training time and accuracy (**C2**). Therefore, an adaptive algorithm that works on diverse datasets is essential.

4 ECHO: Subgraph-Wise Sampling with Correction

To tackle the challenges outlined, we propose a subgraph-wise sampling based training method called ECHO. ECHO consists of two innovations for both the preprocessing and training stages as shown in Fig. 2. In the preprocessing stage, we devise a variance-aware hyperparameter search algorithm (VHS) for training subgraph generation strategy. For specific hyperparameters, VHS simulates the training batch nodes and checks conditions designed for graph drop variance and skewed sample variance. In this way, VHS evaluates the efficiency of subgraph generation strategies, thereby addressing **C1**. In the training stage, we develop an adaptive correction algorithm called AES to address **C2**. AES uses approximate training loss to guide training switching between subgraph-wise sampling epochs and correction epochs. AES adaptively adjusts the intervals between correction epochs on different datasets.

Besides, the correction during the training stage may be insufficient to eliminate the negative effects of large SS variance caused by improper training sub-

graph generation strategy. Both components are essential for fast training and high accuracy.

4.1 Preprocessing Stage of ECHO

In GNN training using subgraph-wise sampling, each mini-batch samples s partitions from the p partitions that were generated during preprocessing. The training subgraph is formed by the nodes in the sampled partitions and the corresponding induced subgraph. In what follows, we analyze the relationship between training subgraphs and variances, including graph drop variance and skewed sample variance. Based on this, we develop conditions for hyperparameter examination to ensure relative low level variance. By grid search with a specific step size, VHS chooses the appropriate hyperparameters for individual datasets.

Variance and Training Subgraphs.

Graph Drop Variance. The graph drop variance is affected by the remaining topological structures in the training subgraphs. Intuitively, if the majority of nodes and their neighbors (within restricted hops) are located in the same partition, the training subgraphs tend to be dense and retain meaningful information. Since the number of training nodes can be unbalanced among partitions, some batches may contain few or even no training nodes, and are therefore skipped to avoid redundancy. The ratio of unskipped batches $r = \frac{|\mathcal{B}_{\text{unskipped}}|}{|\mathcal{B}|}$ in a single epoch directly reflects the fraction of the original graph that is covered during training. Therefore, we estimate the level of graph drop variance through r and propose the condition **M1**: $r \geq \mu$ as an acceptable boundary for structural information loss.

Skewed Sample Variance. In the implementation for an epoch’s training, the partitions are shuffled first, then an identical number (*i.e.*, s) of partitions are selected sequentially for each mini-batch. This leads to insufficient shuffling of training nodes, resulting in the difference between batch label distribution and global label distribution. We quantify such difference by the average during one epoch as $\epsilon = \frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}} |D_{\mathcal{V}_T} - D_B|$ to reflect the level of skewed sample variance, where $D_{\mathcal{V}_T}$ denote the global label distribution, D_B denote the distribution in batch B . Suppose b is batch size, n is sample size in machine learning tasks, Meng *et al.* [14] have demonstrated that if the distribution difference caused by insufficient shuffling is less than $\frac{\sqrt{b}}{n}$, the convergence is not influenced. Motivated by this, we propose the condition **M2**: $\epsilon \leq \nu \sqrt{\frac{s}{pN_t}}$, where $N_t = |\mathcal{V}_t|$ is the sample size, $\frac{N_t \cdot s}{p}$ is an approximation of batch size in SS. **M2** leverages the relationship between skewed sample variance and label distribution in actual training subgraphs.

Variance-Aware Hyperparameter Search (VHS). Since SS batches only contain nodes in the selected partitions, obtaining node partitions enables simulation of actual node sets during training. This allows us to approximate

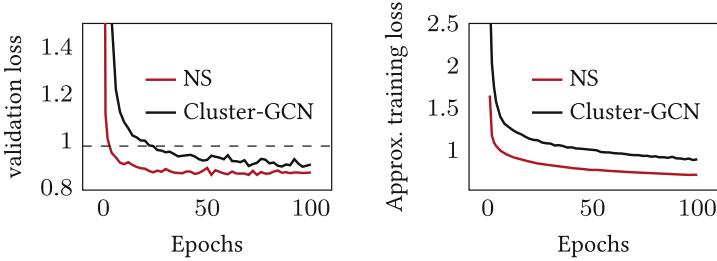


Fig. 3. Comparison of loss with Cluster-GCN and neighbor sampling (NS)

unskipped batch rates r and simulate shuffling error ϵ . To enable lightweight condition checks, we adopt continuous partitioning [22], which logically partitions the node set.

Hyperparameter Search. The search space is defined as $s \geq 1, p \leq |V|, s, p \in \mathbb{N}$ with constant $\frac{p}{s}$. For each pair of p and s , we simulate node sets of mini-batches in one epoch by shuffling logical partitions. Then, the training node label distribution in the simulated data is used to compute the corresponding ϵ and r . Finally, we check whether both conditions **M1** and **M2** hold. Intuitively, as the granularity of shuffling and sampling becomes finer with larger s and p , the introduced variance tends to decrease and is reflected by r and ϵ . Therefore, VHS starts with the option $s = 1, p = p_0$ for computation efficiency of sampling and expands them in the same proportion with a fixed step size during the search. In this way, VHS guarantees a relatively low level of variance when training with the selected hyperparameters.

4.2 Training Stage of ECHO

Correction for Subgraph-Wise Sampling. The idea of correction [5, 17] is to periodically execute training epochs with access to the original graph structure to refine the learned model. Since subgraph-wise sampling restricts structural information confined to partitions, training convergence can be improved by inserting correction epochs to break the restriction of partitions. For efficiency and scalability, ECHO adopts neighbor sampling, which has access to full neighbors, as correction epochs.

Observation on Loss. We compare the decreasing trend of loss for training with subgraph-wise sampling and neighbor sampling in Fig. 3 and obtain two observations. **O1:** subgraph-wise sampling significantly slows down the convergence at early training. Taking training with Cluster-GCN on Ogbn-arxiv [11] as an example, training with neighbor sampling achieves around 96% loss drop after the first 3 epochs, while training with Cluster-GCN needs 24 epochs to reach a similar validation loss. **O2:** the approximate training loss of subgraph-wise sampling is larger than that of neighbor sampling. Besides, the gap tends

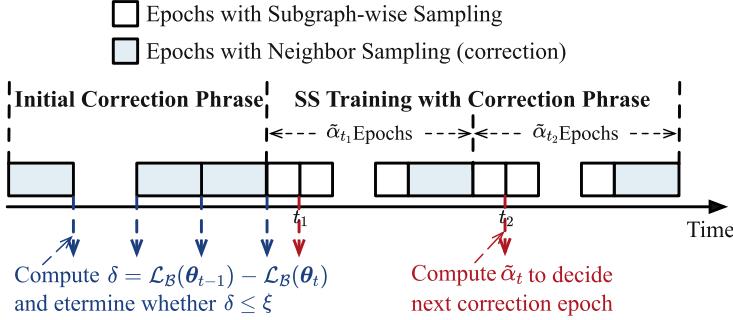


Fig. 4. Training process with AES

to fluctuate as training proceeds as shown on the right of Fig. 3. Based on these observations, we design an adaptive correction algorithm *AES*.

Adaptive Epoch Switch (AES). AES uses approximate training loss to guide training switch between subgraph-wise sampling (SS) epochs and correction epochs that use neighbor sampling, as shown in Fig. 4. The training consists of two phrases: the initial correction phrase and SS training with correction phrase. As implied by **O1**, the early correction epochs are essential to accelerate convergence. Therefore, we execute correction epochs in the initial correction phrase. To determine the end of the initial phase, we set a small threshold of ξ for the drop of approximate training loss between consecutive epochs. Once the approximate training loss drop is less than ξ , ECHO enters the second phrase.

Let \mathcal{L}_P and \mathcal{L}_B denote the approximate training loss of SS epochs and correction epochs, respectively. Inspired by **O2**, we define $\alpha_t = \frac{\mathcal{L}_P(\theta_t)}{\mathcal{L}_B(\theta_t)}$ at epoch t to measure the slow-down of the convergence speed of SS. α_t reflects the impact of SS on convergence, by computing the ratio of the loss downward space for SS and correction under θ_{t+1} . Since the approximate training loss drop of neighbor sampling is less than ξ in the second phrase, we use $\mathcal{L}_B(\theta_{t-1})$ to approximate $\mathcal{L}_B(\theta_t)$ to avoid extra cost. Then we have $\tilde{\alpha}_t = \frac{\mathcal{L}_P(\theta_t)}{\mathcal{L}_B(\theta_{t-1})}$.

In the SS training with correction phrase, every time correction is executed, ECHO executes one SS epoch and calculates $\tilde{\alpha}_t$ to determine the timing of next correction. Between correction epochs, ECHO executes SS epochs. This periodical strategy helps the training to access the global information to correct the learned models, which accelerates the convergence at the same time.

4.3 Theoretical Analysis

In this section, we provide the convergence analysis of ECHO. We first make standard assumption for mini-batch training with neighbor sampling as [17]:

Assumption 3. *The stochastic gradient for mini-batch training with neighbor sampling has stochastic gradient bias bounded by σ_{bias}^2 , i.e.,*

$$\mathbb{E} \left[\mathbb{E} \left[\tilde{\nabla} \mathcal{L}_B(\theta_t) \right] - \nabla \mathcal{L}(\theta_t) \right] \leq \sigma_{bias}^2.$$

We use $\tilde{\nabla} \mathcal{L}_B$ to denote the stochastic gradient of mini-batch training with neighbor sampling. Following the definition of κ_1^2 and κ_2^2 , we analyze the expectation of the stochastic gradient $\nabla \mathcal{L}(\theta_t)$ after T epochs.

Theorem 2 (*Convergence of ECHO*). *With Assumption 1, 2 and 3 hold, let \mathcal{T}_c and \mathcal{T}_s to denote correction epochs and SS epochs executed. Let $G_s = \min_{t \in \mathcal{T}_s} \mathbb{E} \left[\|\nabla \mathcal{L}_P(\theta_t)\|^2 \right]$, $G_c = \min_{t \in \mathcal{T}_c} \mathbb{E} \left[\left\| \tilde{\nabla} \mathcal{L}_B(\theta_t) \right\|^2 \right]$. If we choose the learning rate $\eta = \gamma = \frac{1}{\sqrt{T}}$, such that $|\mathcal{T}_c| \geq \frac{T((L_f \eta - 1)G_s + 2\kappa_1^2 + 4\kappa_2^2)}{(1 - L_f \gamma)G_c - (1 - L_f \eta)G_s + 2\kappa_1^2 + 4\kappa_2^2 - \sigma_{bias}^2}$, then we have :*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \left[\|\nabla \mathcal{L}(\theta_t)\|^2 \right] = \mathcal{O} \left(\frac{1}{\sqrt{T}} \right) \quad (3)$$

Proof. The details on the proof of Theorem 2 is provided in supplementary materials.

Theorem 2 implies that, with carefully chosen learning rates and enough correction epochs, for a specific T , ECHO has the norm of gradient bounded by $\mathcal{O}(\frac{1}{\sqrt{T}})$. That said, ECHO is no longer affected $\mathcal{O}(\kappa_1^2 + \kappa_2^2)$ with enough correction epochs if the condition holds. Under the same T , more correction epochs seem more likely to satisfy the condition in Theorem 2, but cost more time. This is because the correction epochs require more time than that of subgraph-wise sampling epochs. In ECHO, AES provides an adaptive solution to achieve a trade-off in training time and accuracy by incorporating training process insights into correction strategy.

5 Experiments

5.1 Experimental Setup

The experiments evaluate two popular GNN architectures: SAGE [10] and GAT [19], where the parameters are set following [10, 19]. We choose 4 representative sampling-based methods as baselines. Training with neighbor sampling (NS) [10] reaches state-of-the-art accuracy for most datasets. Cluster-GCN (CLUSTER) [3], GraphSAINT (SAINT) [21] and GAS [8] are three representative SS-based methods in literature.

The experiments are performed over four real-world classification datasets: Flickr [21], Reddit [10], Ogbn-arxiv (Arxiv) [11], and Ogbn-products (Product) [11]. These datasets contain thousands to millions of nodes and edges, and have distinct statistical characteristics. Details of datasets and model settings

are reported in supplementary material. We run all experiments on a server with Intel(R) Xeon(R) Silver 4214 CPU (126 GB) and a NVIDIA GeForce RTX 3080 Ti GPU (12 GB). All methods are tested with Torch 1.10.0 and PyG 2.0.4.

We focus on two metrics: final test accuracy and convergence. The final test accuracy is the accuracy on test set using the model that achieves the best accuracy on evaluation set, while convergence is measured by the number of epochs to reach final test accuracy.

5.2 Performance Evaluation

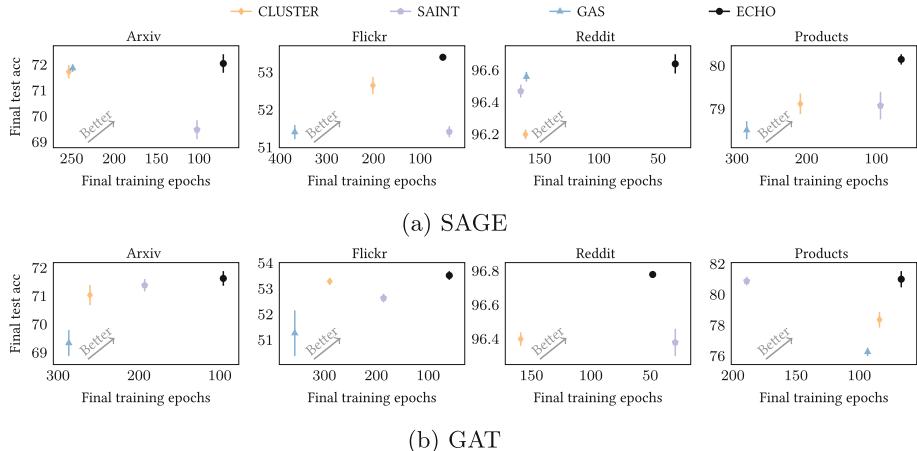


Fig. 5. Performance comparison results on different datasets regarding final test accuracy and convergence denoted as final training epochs

Comparison with Subgraph-Wise Sampling Based Baselines

Convergence and Test Accuracy. Figure 5 plots the test accuracy versus the number of epochs till convergence. ECHO achieves the best test accuracy across all datasets on both SAGE and GAT, and performs the least or second least in terms of epochs. This is partly due to VHS customizing hyperparameter search for specific datasets and partly due to the loss-based guidance of AES that effectively mitigates the negative impact of variance accounting on training progress. ECHO consistently achieves the best trade-off between accuracy and convergence across all cases. In contrast, other SS baselines either show a significant accuracy gap (over 1%) or needs more epochs (up to 11 \times than ECHO) to reach acceptable test accuracy.

Delving into the Training Process. We further plot the validation loss over training epochs on tested datasets in Fig. 6. We use the full graph structure to compute validation loss for all methods. We can see that ECHO exhibits faster decrease in validation loss than other SS-based methods. In contrast, the impact

Table 1. The time (s) to simulate node distribution with subgraph generation algorithms: continuous partitioning, Metis, and random walk sampler.

Datasets	Continuous Partitioning			Metis			Random Walk Sampler		
	p	s	time	p	s	time	batch size	walk length	time
Flickr	4	2	0.0029	24	12	1.26	6000	2	0.0072
Arxiv	10	1	0.0048	500	50	4.04	3000	4	0.0162
Reddit	10	1	0.0052	1500	20	268.75	2000	4	0.0564
Products	1000	50	0.0303	15000	200	283.95	20000	3	0.1280

of subgraph-wise sampling with other SS baselines show more variation on convergence over different datasets due to the varying variances in different datasets. *Preprocessing Time.* In ECHO, VHS provides an efficient solution for the hyperparameter search. Since VHS requires node distribution simulation during training to assess the appropriateness of hyperparameters, we compare such costs for continuous partitioning (used in ECHO) against Metis (used in CLUSTER and GAS) and random walk sampler (used in SAINT) and report in Table 1. Metis is time-consuming, especially when dealing with large datasets, while random walk takes less than 1 s. In contrast, the utilized continuous partitioning takes less than 0.1 s for all datasets, signifying its superior efficiency. Besides, VHS searches proper hyperparameters for specific datasets in preprocessing, lowering the tuning cost of applying to new datasets.

Comparison with Node-Wise Sampling. We finally compare ECHO with node-wise neighbor sampling (NS). The results are listed in Table 2. We can see that the design of ECHO fits our purpose: it achieves comparable or better accuracy while greatly saving training time, especially in large graphs (up to $3.5\times$). Note that the reduction of training time comes from the fact that SS-based methods (including ECHO) take much less time in each training epoch than NS as evidenced in [3, 8, 21]. Nevertheless, SS-based baselines fail to achieve comparable accuracy than NS and fast training at the same time.

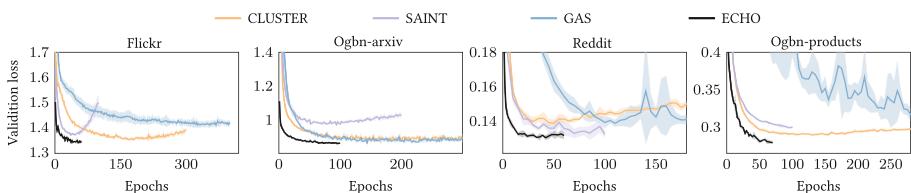


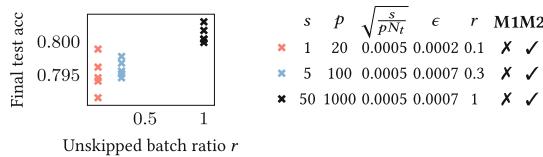
Fig. 6. Validation loss over training epochs on SAGE

Table 2. Training time (s), final training epochs (#E), and final test accuracy (acc.) of ECHO and NS

Models	Datasets	NS			ECHO		
		time	#E	acc.	time	#E	acc.
SAGE	Flickr	100	32	53.37	83	51	53.40
	Arxiv	221	76	71.97	112	71	72.05
	Reddit	1407	76	96.70	451	35	96.64
	Products	1482	80	79.24	702	64	80.15
GAT	Flickr	97	34	53.69	138	59	53.52
	Arxiv	336	90	72.10	230	96	71.63
	Reddit	2360	75	96.76	676	48	96.78
	Products	2854	86	78.83	1134	67	81.01

5.3 Ablation

Effectiveness of Variance-Aware Hyperparameter Search (VHS). For **M1** ($r \geq 0.9$), we conduct training on Ogbn-products with hyperparameters that fulfill only **M1**. Figure 7 displays the final test accuracy and the corresponding r of five runs for each pair of s, p . We can see that the test accuracy decreases with a smaller value of r and tends to be more unstable. The performance drop is primarily due to the significant loss of structural information, resulting in large graph drop variance. For **M2** ($\epsilon \leq \sqrt{\frac{s}{pN_t}}$), we conduct training with hyperparameters that meet **M1** and not meet **M2**. As shown in Table 3, the accuracy on three out of four datasets decreases remarkably. The largest accuracy gap is 0.44%. This demonstrates that ECHO benefits from satisfying **M2** which reflects the level of skewed sample variance.

**Fig. 7.** Final test accuracy and unskipped batch ratio of SAGE on Ogbn-products under different hyperparameters that meet **M2**

Effectiveness of Adaptive Epoch Switch (AES). We further compare AES with other correction strategies in previous works [5, 17]. Past strategies can be formulated with a basic correction interval K and the interval increment coefficient ρ . Then, the internal epochs between two correction is $K\rho^{ct}$. If $\rho = 1$, correction is executed every $K + 1$ epochs. This is the strategy in [5]. If $\rho = 1.1$,

Table 3. Final test accuracy of SAGE under different hyperparameters that meet **M1** and not meet **M2**

Datasets	s	p	ϵ	$\sqrt{\frac{s}{pN_t}}$	M2	Accuracy
Flickr	1	5	0.0075	0.0021	✗	52.96 ↓
Arxiv	1	30	0.0020	0.0006	✗	71.95 ↓
Reddit	1	30	0.0018	0.0005	✗	96.73 ↑
Products	30	1500	0.0011	0.0003	✗	80.07 ↓

the frequency of correction decreases as training proceeds. This is the strategy in [17]. We refer to the settings in [5, 17], and select 4 pairs of K and ρ to evaluate. We name them **S1**: $K = 1, \rho = 1.1$; **S2**: $K = 2, \rho = 1.1$; **S3**: $K = 4, \rho = 1$; **S4**: $K = 9, \rho = 1$.

Figure 8 displays the variance in test accuracy over time for all correction strategies. In particular, we mark the point with final test accuracy. As Fig. 8 shows, ECHO achieves nearly the best trade-off between accuracy and time efficiency among the tested strategies, while other strategies perform differently on distinct datasets. This benefits from the guidance of approximate training loss, which provides a view for training process. Besides, we show the validation loss over training epochs in Fig. 9. AES shows the fastest convergence among all five correction strategies. Other strategies may execute inadequate corrections at certain epochs, thus slow down the convergence speed.

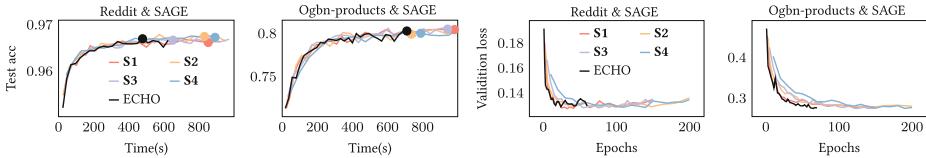


Fig. 8. Test accuracy over time with different correction strategies

Fig. 9. Validation loss over training epochs with different correction strategies

6 Conclusion

In this paper, we decouple the variance of mini-bath training with subgraph-wise sampling into two components: the graph drop variance and skewed sample variance. These variances cause an irreducible residual error that slows down the convergence of subgraph-wise sampling. To reduce such variances, we propose ECHO that works at both the preprocessing and training stages. ECHO adopts lightweight continuous partitioning for efficient node distribution simulation, and proposes conditions for variance-aware hyperparameter search. It further uses an adaptive epoch switch algorithm to allow the training access to the original input graph when needed. We show through analysis that ECHO

attains $\mathcal{O}(1/\sqrt{T})$ convergence, where T is the number of training epochs. Our experiments demonstrate that ECHO achieves up to $3.5 \times$ training time speed-up compared to node-wise sampling and up to $11 \times$ convergence speed-up compared to SS baselines.

In the future, we plan to investigate other correction techniques (*e.g.*, introducing edge drop technology), in order to further accelerate the training. Our theoretical analysis and the proposed algorithm of epoch switch pave the way for the design of more efficient and flexible correction solutions.

Acknowledgments. This work is supported in part by National Key R&D Program of China (Grant No. 2022YFB3103000), in part by the National Natural Science Foundation of China (Grant No. U20A20180).

References

1. Boldi, P., Vigna, S.: The WebGraph framework I: compression techniques. In: Proceedings of the 13th International Conference on World Wide Web, pp. 595–602 (2004)
2. Chen, J., Ma, T., Xiao, C.: FastGCN: fast learning with graph convolutional networks via importance sampling. In: International Conference on Learning Representations (2018)
3. Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J.: Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 257–266 (2019)
4. Cong, W., Forsati, R., Kandemir, M., Mahdavi, M.: Minimal variance sampling with provable guarantees for fast training of graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1393–1403. Association for Computing Machinery, New York, NY, USA (2020)
5. Cong, W., Ramezani, M., Mahdavi, M.: On the importance of sampling in learning graph convolutional networks. arXiv preprint [arXiv:2103.02696](https://arxiv.org/abs/2103.02696) (2021)
6. Dong, J., Zheng, D., Yang, L.F., Karypis, G.: Global neighbor sampling for mixed CPU-GPU training on giant graphs. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 289–299. Association for Computing Machinery, New York, NY, USA (2021)
7. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
8. Fey, M., Lenssen, J.E., Weichert, F., Leskovec, J.: GNNAutoScale: scalable and expressive graph neural networks via historical embeddings. In: Proceedings of the 38th International Conference on Machine Learning, pp. 3294–3304. PMLR, 18–24 July 2021
9. Hamaguchi, T., Oiwa, H., Shimbo, M., Matsumoto, Y.: Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI, pp. 1802–1808 (2017)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2017)

11. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. In: Advances in Neural Information Processing Systems, pp. 22118–22133. Curran Associates, Inc. (2020)
12. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **1**, 359–392 (1998)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
14. Meng, Q., Chen, W., Wang, Y., Ma, Z.M., Liu, T.Y.: Convergence analysis of distributed stochastic gradient descent with shuffling. Neurocomputing, 46–57 (2019)
15. Pal, A., Eksombatchai, C., Zhou, Y., Zhao, B., Rosenberg, C., Leskovec, J.: PinnerSage: multi-modal user embedding framework for recommendations at Pinterest. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2311–2320. Association for Computing Machinery, New York, NY, USA (2020)
16. Park, N., Kan, A., Dong, X.L., Zhao, T., Faloutsos, C.: Estimating node importance in knowledge graphs using graph neural networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 596–606. Association for Computing Machinery, New York, NY, USA (2019)
17. Ramezani, M., Cong, W., Mahdavi, M., Kandemir, M., Sivasubramaniam, A.: Learn locally, correct globally: a distributed algorithm for training graph neural networks. In: International Conference on Learning Representations (2022)
18. Stokes, J.M., et al.: A deep learning approach to antibiotic discovery. Cell **4**, 688–702 (2020)
19. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
20. Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., Tang, J.: GeoDiff: a geometric diffusion model for molecular conformation generation. In: International Conference on Learning Representations (2022)
21. Zeng, H., Zhou, H., Srivastava, A., Kannan, R., Prasanna, V.: GraphSAINT: graph sampling based inductive learning method. In: International Conference on Learning Representations (2020)
22. Zhu, X., Chen, W., Zheng, W., Ma, X.: Gemini: a computation-centric distributed graph processing system. In: 12th USENIX Symposium on Operating Systems Design and Implementation, pp. 301–316. USENIX Association, Savannah, GA, November 2016
23. Zou, D., Hu, Z., Wang, Y., Jiang, S., Sun, Y., Gu, Q.: Layer-dependent importance sampling for training deep and large graph convolutional networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2019)



A Parallel and Distributed Data Management Approach for MEC Using the Improved Parameterized Deep Q-Network

Bingqing Ren^{1,2} , Peng Yang^{1,2} , Meng Yi^{1,2} , and Dongmei Yang³

¹ School of Computer Science and Engineering, Southeast University, Nanjing, China
pengyang@seu.edu.cn

² Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China

³ School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

Abstract. With the rapid development of Multi-access Edge Computing (MEC), massive distributed collaborative data are generated in real-time at the edge of the network, which brings great challenges to the parallel and distributed data management for compute-intensive and low-latency application requirements. Traditional offloading strategies often neglect signal interference and the critical interdependence between edge servers. Our work combines the parallel and distributed data management system with the operational requirements of MEC. We employ a Partially Observable Markov Game (POMG) framework alongside an Improved Parameterized Deep Q-Network (I-PDQN) algorithm, specifically designed for complex decision-making scenarios with task offloading and resource allocation in a discrete-continuous hybrid action space. The collaboration between the POMG framework and the I-PDQN algorithm facilitates data management and analysis across multiple edge servers and it addresses the dual challenges of parallel and distributed data management by ensuring data consistency and availability across distributed nodes, while optimizing computational resources to reduce latency and energy consumption. Experimental results illustrate that compared with the start-of-the-art baselines, our approach achieves competitive improvement in energy consumption and time delay.

Keywords: Multi-access Edge Computing · parallel and distributed data management · task offloading and resource allocation · deep reinforcement learning

1 Introduction

In the rapidly developing multi-access Edge Computing (MEC) network system, the rapid generation of real-time distributed and cooperative data has brought

great challenges to processing a large amount of data at the edge of the network and meeting the needs of computation-intensive and delay-sensitive applications. The MEC architecture brings the computing power closer to the data source, and it is expected to significantly reduce the response time and bandwidth usage. However, it simultaneously introduces complexities related to data management, task allocation, and resource allocation. Therefore, edge computing solutions should be innovated and refined to adapt to dynamically changing workloads, manage data efficiently in a distributed manner, and maintain cooperation between different edge servers without compromising performance and security.

In the edge computing network, tasks arrive dynamically, and each of them is characterized by a different specific request. Which edge server the task is offloaded to and how much computing resources are allocated have a great impact on the task completion delay and energy consumption. Particularly in multi-edge server environments, where signal coverage overlaps occur, tasks from users in overlapping regions can be offloaded to one of the edge servers for processing. Inappropriate task offloading and resource allocation not only affect task completion latency but also disrupt load balancing among edge servers, ultimately impacting the overall user capacity of the server cluster [1]. Therefore, it is necessary to design a rational task offloading as well as resource allocation scheme that satisfies the latency requirements of tasks while maximizing number of serviced users and minimizing average energy consumption of all completed tasks.

In the collaborative multi-edge server scenario, a centralized controller is no longer employed to manage individual edge servers. Instead, each edge server operates independently and autonomously, making task offloading as well as resource allocation decisions based on its local observations, which also involves the mixed discrete-continuous action space [2]. As signal coverage overlaps exist between edge servers, the offloading decision of tasks in the overlapping region to one edge server will affect the other. Thus, this scenario constitutes a cooperative game. Due to limited computational capacity of each edge server, the signal coverage is not infinite. And considering dynamic nature of task arrivals as well as departures in both time and space, operating in isolation can lead to problems such as load imbalance. Some edge servers may face resource shortages due to heavy task offloading, while others may be underutilized. The cooperation of multi-edge servers can solve the problem that a single edge server cannot be used due to failure. At the same time, the partial cross coverage of edge server signals can solve the problem that users in blank areas cannot be covered [3]. Above all, a cooperative model is proposed for multi-edge servers, whereby edge servers collaborate to perform task offloading and resource allocation, thereby improving utilization of resources in the MEC network. Our study is set against the backdrop of a decentralized network of edge servers, where the absence of a centralized control mechanism underscores the importance of efficient data management across distributed systems. The contributions of our investigation are enriched by incorporating principles of parallel and distributed computing, delineated as follows:

- We construct a dynamic and cooperative multi-edge server network environment, where each edge server operates independently and autonomously, emphasizing the challenge caused by signal overlap, and applying a parallel and distributed data management architecture to explore the interaction between multiple edge servers.
- We jointly modeled the task offloading and resource allocation problem under multi-edge server cooperation as Partially Observable Markov Game (POMG), and Improved Parameterized Deep Q-Network (I-PDQN) algorithm with the discrete-continuous hybrid action space is used to solve it. While meeting the computing requirements of edge computing, parallel processing enhances server cooperation ability and data throughput.
- Comparative simulations show that the proposed method significantly outperforms standard benchmarks in terms of processing delay and energy efficiency. By utilizing distributed data management technology, the proposed method balances the utilization of computing resources and promotes the cooperation between edge servers to meet the latency requirements, maximize the number of served users and reduce the average energy consumption.

2 Related Work

In the research of task offloading and resource allocation in collaborative multi-edge server environments, experts and scholars from both domestic and international communities have made significant contributions. Lai et al. [4] addressed the edge user allocation issue from perspective of application providers and employed integer linear programming techniques for solving, further utilizing heuristic methods to search for suboptimal solutions. He et al. [5] proposed a cost-effective edge user allocation issue and modeled it as a potential game, designing a decentralized method to determine Nash equilibrium of game, which serves as a solution to edge user allocation problem. Tran et al. [6] considered the scenario with multiple wireless networks, where users first offload tasks and then edge servers assist in task processing, aiming to increase offloading benefits for users. Yu et al. [7] tackled challenge of MEC technologies being unsuitable for scenarios with explosive growth in user numbers or sparse network infrastructures, and presented a solution that involves the collaboration between edge clouds and unmanned aerial vehicles to provide services for IoT devices. Deng et al. [8] investigated the problem of maximizing long-term throughput in multi-user multi-edge server systems, employing Markov decision processes to method queue states of mobile devices and edge servers, and combining matching theory to devise a joint user association as well as resource allocation method. Ale et al. [9] focused on task offloading in dynamic systems with multiple edge servers, aiming to maximize the number of serviced tasks and minimize energy consumption, utilizing reinforcement learning methods for offloading and computation resource allocation.

Zhou et al. [10] investigated task offloading as well as resource allocation problems considering the collaborative relationships among edge servers. They

proposed a two-layer method, where upper-layer method was inspired by evolutionary method to globally search for the optimal offloading scheme, and the lower-layer algorithm aimed to generate an allocation scheme that fully utilizes server resources while considering fairness among all tasks. Zhao et al. [11] jointly designed the flight trajectory of unmanned aerial vehicles, task allocation, and communication resource management to address task offloading problems. They employed a collaborative multi-agent deep reinforcement learning framework to reduce sum of execution latency and energy consumption. Chen et al. [12] proposed a cooperative resource allocation framework for multi-edge server environments to solve complex resource allocation problems. He et al. [13] addressed the peer offloading problem based on a random task arrival pattern. They proposed two online algorithms considering the presence or absence of prior knowledge about arrival rates. Wang et al. [14] considered a three-layer cooperative MEC network involving end devices, edge nodes, and cloud servers. Vertical cooperation was formed among devices, edge nodes, and cloud servers, while horizontal cooperation was established among edge nodes.

Xu et al. [15] addressed challenge of cloud computing's inability to cope with rapid growth of data volume. They proposed a collaborative architecture that combines cloud computing and edge computing. Zhao et al. [16] proposed a cloud-edge collaborative task offloading problem to jointly optimize task offloading and resource allocation decisions. They designed a distributed method to obtain optimal solution. Ho et al. [17] jointly optimized server selection, collaborative offloading, and task handoff decisions to minimize the cost of total latency. Ning et al. [18] considered partial offloading issue of tasks as well as proposed a cloud-edge collaborative task offloading approach. Huang et al. [19] addressed the challenges of interference among edge servers, diverse application requirements, and dynamic wireless environments. They proposed a collaborative offloading strategy to minimize energy consumption costs.

Through above review, it is observed that existing research mainly focuses on static optimization issue of task offloading as well as resource allocation [20, 21]. However, actual MEC environment is complex and dynamic, and task demands are stochastic. Some works consider scenarios with only one edge server in MEC environment [22, 23] and do not take into account mutual influence among edge servers, especially when the signals of edge servers partially overlap. Therefore, this study considers the environment of multiple edge servers to investigate issue of task offloading as well as resource allocation.

3 Partially Observable Markov Game Model

In this paper, we consider a cooperative game model involving multiple edge servers, which is represented by a seven-tuple $\{S, A, r, P, \gamma, T, M\}$, known as a Partially Observable Markov Game (POMG) [24]. In this game, there are M intelligent agents, where each edge server is treated as an individual agent. The global state set S represents the overall state of MEC environment, while o_j denotes the local observation of edge server e_j . Due to limited observability, each

edge server can only observe its own resource usage and is unable to observe the state information of other edge servers or the task information beyond its signal coverage. Therefore, this model is considered partially observable.

At the beginning of t , each user device u in each edge server e_j randomly generates a task, denoted by $\rho_{it} = \{d_{it}, c_{it}, \tau_{it}\}$, where d_{it} , c_{it} , τ_{it} represents the size of task data arriving at time t , total computation resource required for task, and maximum acceptable completion time for the user during task execution. Thus, for edge server e_j , its locally observable state information at time t is represented as $o_{jt} = \{d_{it}, c_{it}, \tau_{it}, l_{ijt}, w_{jt}, b_{j(t-1)}\} \in O_j$, where l_{ijt} represents distance from user u_i to edge server e_j , w_{jt} and $b_{j(t-1)}$ represent the remaining computation resources and the computation resources returned in the previous time step for edge server e_j , respectively.

In this work, all the edge servers share the same frequency band. The total bandwidth of each edge server is denoted by B_w , which is separated into several orthogonal subchannels to avert intra-cell interference. ϕ_{ijt} will denote the uplink signal-to-noise ratio (SNR) from user u_i to edge server e_j at t , which is given by

$$\phi_{ijt} = \frac{p_{ijt}g_{ijt}}{\sigma^2} \quad (1)$$

where p_{ijt} represents the transmission power of user device u_i , g_{ijt} is the channel gain and σ^2 is the variance of the modeled additive white Gaussian noise. Then, the uplink data transmission speed from user device u_i to edge server e_j at t is expressed as

$$v_{ijt} = \log_2(1 + \phi_{ijt}) = \log_2\left(1 + \frac{p_{ijt}g_{ijt}}{\sigma^2}\right) \quad (2)$$

Thus, the uplink transmission latency of task ρ_{it} is $L_{ijt}^{tran} = \frac{d_{it}}{v_{ijt}}$. The computational resources allocated for processing this task by the edge server e_j are defined by $f_{ijt} \in [0, f_j]$. The computing latency of task ρ_{it} on the edge server e_j can be expressed by $L_{ijt}^{comp} = \frac{c_{it}}{f_{ijt}}$. Therefore, the total latency for completing task ϕ_{ijt} is given by

$$L_{ijt} = \frac{d_{it}}{v_{ijt}} + \frac{c_{it}}{f_{ijt}} \quad (3)$$

The energy consumption of the task ρ_{it} transmission to the edge server e_j and processing are given by $E_{ijt}^{tran} = p_{ijt}^{tran}L_{ijt}^{tran} = \frac{p_{ijt}^{tran}d_{it}}{v_{ijt}}$ and $E_{ijt}^{comp} = p_{ijt}^{comp}c_{it}$, where p_{ijt}^{comp} represent the total energy consumed by the edge server e_j per cycle. Therefore, the total energy consumed for completing task ρ_{it} is given by

$$E_{ijt} = \frac{p_{ijt}^{tran}d_{it}}{v_{ijt}} + p_{ijt}^{comp}c_{it} \quad (4)$$

In this paper, $s = (o_1, o_2, \dots, o_M) \in S$ is used to represent a state of the MEC environment. The joint action space $A = A_1 \times \dots \times A_M$ is denoted for M edge servers. At every time step t , each edge server chooses an action

$a_j = (k_j, x_{k_j}) \in A_j$ to execute based on its policy π_j and the local observation value o_j . The collective actions of all edge servers form the joint action $\vec{a} \in A$ after all edge servers make task offloading and resource allocation decisions. k_j represents the task offloading decision made by edge server e_j , which is a discrete value indicating where the task is offloaded for execution. x_{k_j} represents resource allocation decision, which is a continuous value representing the allocated computation resource for the task. After executing the joint action, the state transition probability $P(s_{t+1}|s, \vec{a}) : S \times A \times S \rightarrow [0, 1]$ generates the next state s_{t+1} of the MEC environment. All edge servers share the same reward function $r(s, \vec{a}) : S \times A \rightarrow R$ and obtain the next observation value $o_{j(t+1)}$. The parameter γ represents the discount factor, and T represents the time sequence. By repeating this process, each edge server aims to maximize the global reward $R = \sum_{t=0}^T \gamma^t r$.

The objective of this paper remains to design a reasonable task offloading and resource allocation scheme that satisfies task latency requirements while maximizing number of served users and minimizing average energy consumption of all tasks. Problem addressed in this paper is the task offloading as well as resource allocation issue in context of multi-edge server collaboration, which also involves a mixed discrete-continuous action space.

4 Task Offloading and Resource Allocation Strategy Based on I-PDQN

Task Offloading and Resource Allocation Strategy based on I-PDQN [27] in the Previous Section, the problem of task offloading and resource allocation in the context of multi-edge server collaboration was formulated as a POMG model. In this paper, the multi-agent reinforcement learning algorithm I-PDQN is employed to generate task offloading and resource allocation strategies. Considering the issue of collaboration among edge servers, they no longer operate in isolation but work together to provide services to users. Each edge server has limited computing resources, and its signal coverage is not infinite. Operating in isolation can lead to problems such as load imbalance. We take into consideration a multi-cell, multi-server MEC system like the one shown in Fig. 1, where each BS is fitted with a MEC server to offer compute offloading services to the resource-constrained mobile users such smart phones, tablets, and wearable devices. Generally speaking, each MEC server can be a physical server or a virtual machine with moderate processing capabilities given by the network operator, and it can communicate with mobile devices using wireless channels provided by the corresponding BS. Each mobile user has the option to select one of the local BSs it can connect to and offload computing work to a MEC server from there. We will use the terms MEC servers and BSs interchangeably for the sake of clarity.

In this paper, problem of task offloading as well as resource allocation in the context of multi-edge server collaboration is first formulated as a POMG model. As this problem involves a mixed discrete-continuous action space, it can

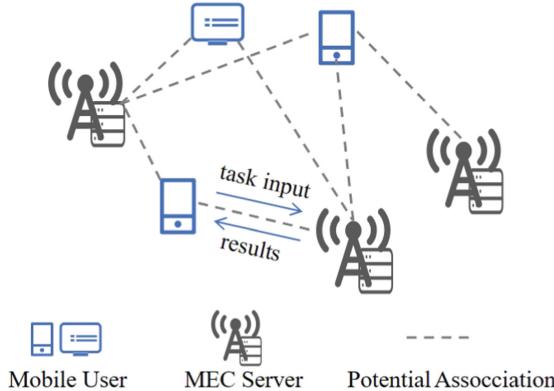


Fig. 1. Example of a cellular system with MEC servers deployed at the BSs

be solved using multi-agent reinforcement learning algorithms. Currently, there has been comprehensive research on multi-agent reinforcement learning algorithms in either discrete action spaces or continuous action spaces, with notable examples being QMIX [25] for discrete action domains and MADDPG [26] for continuous action domains. However, these existing multi-agent algorithms are limited to either discrete or continuous action spaces and cannot handle the mixed discrete-continuous action space encountered in this problem. To address this, Fu et al. [27] extended the P-DQN algorithm to the multi-agent cooperative model, creating the I-PDQN algorithm. This algorithm equips each agent with an independent P-DQN method and employs the paradigm of independent learning. There are N terminal devices and M MEC servers in system model, which is a multiuser multiserver application scenario. For user equipment, the base station serves as a source of communication resources.

Specifically, for each agent, the same setup as the P-DQN algorithm is employed. Considering there are M agents in this game, each agent takes its observation o_j as input, which includes information about task requests and the state of each edge server. Then, a deterministic actor network $\mu_{k_j}(\theta_j)$ is used to output the optimal continuous parameters corresponding to all possible discrete actions of the agent, representing the resource allocation parameters associated with different task offloading options. Subsequently, each agent employs a Q network $Q_j(\omega_j)$ to output action corresponding to maximum Q-value. Finally, optimal mixed action $(k_j^*, x_{k_j}^*)$ for task offloading and resource allocation is formed. This is represented by following equation:

$$(k_j^*, x_{k_j}^*) = \arg \max_{(k_j, x_{k_j})} Q_j(o_j, (k_j, x_{k_j}); \omega_j) \quad (5)$$

where ω_j represents the action Q-network parameters of the agent. In order to achieve the collaborative update between Q-networks of each agent, a mixing network is utilized to produce a fully centralized state-action value function Q_c , so as to promote collaborative update of distributed policies in the hybrid

action space. The mixing network inputs the Q_j of each agent and mixing them monotonically, donated as:

$$Q_c = f(s, Q_1, \dots, Q_M; \omega_{mix}) \quad (6)$$

where f is a non-linear complex function and ω_{mix} is the mixing network weights, which can be updated along with ω_j by minimizing the loss:

$$L(\omega) = \mathbb{E}_{s_t, \vec{k}, \vec{x}_k, r, s_{t+1} \sim \mathcal{D}} [y_c - Q_c(s_t, \vec{k}, \vec{x}_k)]^2 \quad (7)$$

$$y_c = r + \gamma \max_{\vec{k}', \vec{x}_{k'}} Q_c(s', \vec{k}', \vec{x}_{k'}(\theta')) \quad (8)$$

where θ' are parameters of target policy networks. Finally we update the deterministic actor networks $\mu_{k_j}(\theta_j)$ for each agent by maximizing Q_c^s , the gradient can be given as:

$$\nabla_{\theta_j} l(\theta_j) = \mathbb{E}_{s, \vec{k} \sim \mathcal{D}} [\nabla_{\theta_j} \mu_{k_j}(o_j) \nabla_{x_{k_j}} Q_c^s(s, \vec{k}, \vec{x}_k; \omega)] \quad (9)$$

$$Q_c^s = f(s, \hat{Q}_1, \dots, \hat{Q}_M; \omega_{mix}) \quad (10)$$

where $\hat{Q}_j = \sum_{k_j=1}^{K_j} Q(o_j, k_j, x_{k_j}; \omega_j)$, $k_j \in \{1, 2, \dots, K_j\}$. Pseudocode for task offloading as well as resource allocation strategy based on the I-PDQN method is shown in Algorithm 1.

5 Experimental Setup

In this section, we will set experimental parameters and present comparative algorithms used in this study.

5.1 Experimental Parameter Setting

This paper considers MEC environment under cooperation of multiple edge servers. In this collaborative MEC environment, edge servers no longer operate in isolation and there is signal overlap between them. Suppose there are $M = 3$ edge servers, and the signal coverage radius is $\beta = 145$ m. Geographic coordinates of the edge servers are $(100, 100)$, $(200, 200)$, and $(100, 300)$, respectively. Since each edge server has limited resources, the bandwidth and computing capacity of each edge server are set to $B = 15$ MHz and $F = 6$ GHz, respectively. The parameter settings mentioned in the previous context remain the same. Considering that the computing capacity of edge servers is typically 2 to 5 times that of user mobile devices, maximum computing capacity of user mobile devices is set as $fl_{max} = 2.4$ GHz. The total length of time slots is set to $T = 100$. At each time t , tasks dynamically arrive at random locations. For users in signal overlap regions, their tasks can be offloaded to one of the edge servers for processing or executed on the user mobile devices. Suppose the data size of each task follows a uniform distribution $d_i \sim U(8000, 10000)$ kbytes, and the required CPU

Algorithm 1. Task Offloading and Resource Allocation Strategy based on I-PDQN

Input: The position coordinates of each edge server p_j^e ; task request information H ; and the user position coordinates p_i^u ;

Output: Optimal joint TO and RA strategies π_j for each edge server;

- 1: Initialize sample size φ for gradient descent;
- 2: Initialize network weights ω_j and θ_j randomly;
- 3: **for** $i = 1, 2, \dots, \Gamma$ **do**
- 4: $o_{jt} \leftarrow (d_{it}, c_{it}, \tau_{it}, l_{ijt}, w_{jt}, b_{j(t-1)})$;
- 5: **for** $t = 1, 2, \dots, T$ **do**
- 6: $x_{k_j} \leftarrow \mu_{k_j}(o_{it}; \theta_j) + \mathcal{N}$;
- 7: **if** $rnd < \epsilon$ **then**
- 8: $a_{jt} \leftarrow$ random discrete actions;
- 9: **else**
- 10: $a_{jt} \leftarrow (k_{jt}^*, x_{k_{jt}}^*)$ via (5);
- 11: **end if**
- 12: $\vec{a} \leftarrow A_1 \times \dots \times A_M \in A, R \leftarrow \sum_{t=0}^T \gamma^t r$;
- 13: the next state s_{t+1} with $P(s_{t+1}|s, \vec{a}) : S \times A \times S \leftarrow [0, 1]$;
- 14: save $(o_{jt}, a_{jt}, r_t, o_{j(t+1)})$ in D_j ;
- 15: sample φ random $(o_{jt}, a_{jt}, r_t, o_{j(t+1)})$ from D_j
- 16: update Q networks by minimizing the loss function (7);
- 17: update deterministic actor networks via (9);
- 18: **end for**
- 19: **end for**

cycles for tasks follow a uniform distribution $c_i \sim U(6500, 8500)$ MHz. During task execution, certain transmission and execution time is required. Considering that different users have different requirements for task completion time, the maximum acceptable completion time for users follows a uniform distribution $\tau \sim U(2, 4)$ s. The transmission power from user mobile devices to edge servers is set as $p^{trans} = 0.5$ W, the channel gain power is set as $h = 1.02e^{-13}$ W, and the background noise power is set as $\sigma^2 = 1e^{-13}$ W. The energy consumption coefficients for user mobile devices and edge servers are set as $k_u = 1e^{-27}$ and $k_s = 1e^{-29}$, respectively.

In the MEC environment under the cooperation of multiple edge servers, each edge server is an independent and autonomous agent. Each agent can obtain the local state information of the environment through observations and make offloading and resource allocation decisions for dynamically arriving tasks based on this information. The agents collaborate with each other to meet the task latency requirements, maximize the number of served users, and minimize the average energy consumption of all tasks. Similarly, the problem in this paper involves a mixed discrete-continuous action space and collaboration among agents. Considering the nature of the problem, the I-PDQN algorithm is used to generate task offloading and resource allocation strategies. In the I-PDQN algorithm, the Actor network for computing continuous parameters and the Q network for outputting optimal discrete actions both use a two-layer neu-

ral network model. The number of neurons in each layer is set to 128 and 64, respectively. Both the Actor network and Q network use the ReLU activation function. The discount factor γ is set to 0.95, and the soft update coefficients α_1 and α_2 are both set to 0.001. The learning rates for the Actor network and Q network are set to 0.0001 and 0.00001, respectively, and both networks use the Adam optimizer to update the network weights. The size of the experience replay buffer is set to 100,000, and the number of samples φ used for gradient descent is set to 128. The initial values for the parameters ω and θ are set to 0.5. The maximum number of iterations for the experiments is set to 5000.

All the experiments in this paper were conducted on Huawei Cloud servers with the following configuration details: The CPU model is Intel(R) Xeon(R) Gold 6278C CPU @ 2.60 GHz, with 8 cores. The GPU model is NVIDIA-SMI 418.39, with 15079 MB of memory. The CUDA version is 10.1. The operating system information is Linux ecs-a6c1-0001 4.15.0-91-generic.

5.2 Comparison Algorithms

The following algorithms were adapted and compared in the context of multi-edge server cooperation for task offloading and resource allocation in this study: Random algorithm, Greedy algorithm, NO (Nearest offloading) algorithm, and HTR [25] algorithm. The objective is to maximize the number of served users and minimize the average energy consumption of completed tasks subject to the delay requirements of the tasks. The performance of the algorithms is evaluated based on several experimental metrics, including the total number of serviced users, average energy consumption, and average service time.

6 Experimental Analysis

In this paper, each edge server acts as an independent agent, and they are influenced by each other due to the cross-coverage of signals. The decision made by one edge server will have an impact on the neighboring edge servers, making it a cooperative problem. The task offloading and resource allocation problem is formulated as a POMG model, and the I-PDQN algorithm is used for solving it. The research objective is to maximize the number of served users while meeting the task latency requirements and minimizing the average energy consumption of all completed tasks.

6.1 Impact of User Quantity on Experimental Metrics

In the first group of experiments, the user quantity N was varied as $N = \{50, 100, 150, 200\}$ to observe the experimental results, as shown in Fig. 2. As the user quantity increased, the total number of served users generally increased. From the graph, it can be observed that the I-PDQN algorithm achieved the highest total number of served users, while the Random algorithm had the lowest total number of served users. This is because the Random algorithm makes

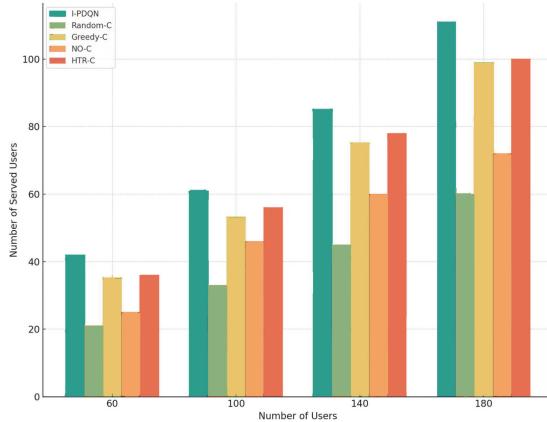


Fig. 2. Number of users vs total number of served users

random decisions for task offloading and resource allocation, which may not effectively meet the timing requirements of more tasks. The HTR algorithm and Greedy algorithm achieved similar total numbers of served users. This is because both algorithms prioritize the use of computing resources from edge servers to serve users. Moreover, the considered tasks in this study are computationally intensive and sensitive to timing requirements. As a result, each edge server can only serve a limited number of tasks in a given time period. Therefore, the total numbers of served users for these two algorithms are approximately equal.

6.2 Impact of Edge Server Computing Performance on Experimental Metrics

In the second set of experiments, the impact of edge server computing performance on the experimental metrics was observed by varying the computing performance of the edge servers, as shown in Fig. 3. It can be observed that as the edge server computing performance increases, the total number of served users also increases. When the edge server computing performance increases from 6 GHz to 7 GHz, the total number of served users grows faster compared to the range of 4 GHz to 6 GHz. This is because higher computing performance reduces the time required to process tasks, ensuring the fulfillment of task latency requirements and successful task completion. Consequently, the total number of served users by the edge servers increases. In addition, the average energy consumption of the I-PDQN algorithm is approximately equal to that of the HTR algorithm. However, the I-PDQN algorithm promotes collaboration among edge servers for task offloading and resource allocation, balancing the utilization of computing resources between user mobile devices and edge servers. As a result, the I-PDQN algorithm outperforms the HTR algorithm in terms of the total number of served users.

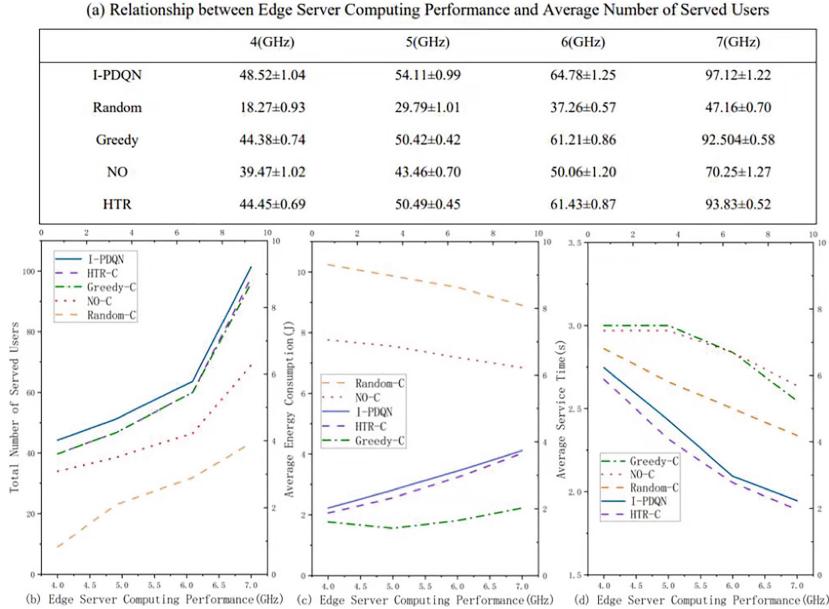


Fig. 3. Impact of edge server computing performance on experimental metrics

6.3 Impact of Edge Server Bandwidth on Experimental Metrics

In the third set of experiments, we investigated the impact of edge server bandwidth on the experimental metrics by varying the bandwidth parameter and the results are shown in Fig. 4. It can be observed that as the edge server bandwidth increases, the average energy consumption decreases. This is because a higher bandwidth reduces the transmission delay and subsequently reduces the energy consumption required for task transmission to the edge servers. The Greedy algorithm aims to minimize energy consumption by making task offloading and resource allocation decisions, resulting in lower average energy consumption. The NO algorithm has lower average energy consumption compared to the Random algorithm because it can select the nearest edge server for task offloading, and executing tasks on edge servers consumes less energy compared to executing them on user mobile devices. Additionally, since the task offloading decisions of the Random algorithm are random, its average energy consumption is higher than that of the NO algorithm. The HTR algorithm aims to minimize the sum of delay and energy consumption in making task decisions. However, the I-PDQN algorithm can balance the usage of computing resources between user devices and edge servers based on task characteristics and the state of edge server computing resources. It also promotes collaboration among edge servers to maximize the number of served users and minimize energy consumption. Therefore, the I-PDQN algorithm serves a slightly higher number of users compared to the HTR algorithm Fig. 4.

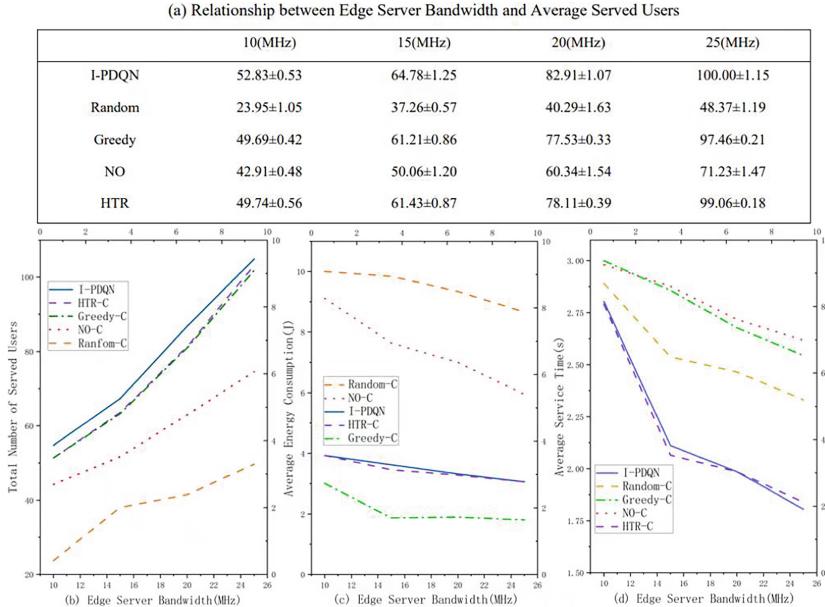


Fig. 4. Impact of edge server bandwidth on experimental metrics

7 Conclusion

In multi-access Edge Computing (MEC) systems, the generation of large amounts of real-time distributed and collaborative data at the edge of the network brings challenges to meet the requirements of computation-intensive and delay-sensitive applications. This paper studies the problem of task offloading and resource allocation in a collaborative multi-edge server environment. Since the problem involved discrete-continuous hybrid action Spaces, it was first modeled as a POMG model. Then, the multi-agent reinforcement learning I-PDQN algorithm was used to solve the problem. Enabling it to make intelligent offloading decisions, optimize energy consumption, and ensure that computational demands are met with minimal latency. In order to evaluate the performance of the algorithm, a number of experimental parameters and indicators were changed to conduct experiments and compared with other four algorithms. Experimental results show that the task offloading and resource allocation strategies generated by the I-PDQN algorithm effectively increase the total number of users served under different parameter Settings. The balance between the utilization of computing resources of edge servers and user mobile devices is achieved. The collaboration between edge servers is promoted to meet the delay requirements of more tasks. It maximizes the number of users served and reduces the average energy consumption to complete tasks.

Acknowledgement. This work was supported in part by the Consulting Project of Chinese Academy of Engineering under Grant 2023-XY-09, the National Natural Science Foundation of China under Grant 62272100, and in part by the Academy-Locality Cooperation Project of Chinese Academy of Engineering under Grant JS2021ZT05.

References

1. Chen, Y., Zhao, J., Wu, Y., Huang, J., Shen, X.S.: QoE-aware decentralized task offloading and resource allocation for end-edge-cloud systems: a game-theoretical approach. *IEEE Trans. Mob. Comput.* **23**(1), 769–784 (2022)
2. Guo, H., Zhou, X., Wang, J., Liu, J., Benslimane, A.: Intelligent task offloading and resource allocation in digital twin based aerial computing networks. *IEEE J. Sel. Areas Commun.* **41**(10), 3095–3110 (2023)
3. Cong, Y., Xue, K., Wang, C., Sun, W., Sun, S., Hu, F.: Latency-energy joint optimization for task offloading and resource allocation in MEC-assisted vehicular networks. *IEEE Trans. Veh. Technol.* **72**(12), 16369–16381 (2023)
4. Lai, P., He, Q., Cui, G., et al.: QoE-aware user allocation in edge computing systems with dynamic QoS. *Futur. Gener. Comput. Syst.* **112**, 684–694 (2020)
5. He, Q., Cui, G., Zhang, X., et al.: A game-theoretical approach for user allocation in edge computing environment. *IEEE Trans. Parallel Distrib. Syst.* **31**(3), 515–529 (2019)
6. Tran, T.X., Pompili, D.: Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Trans. Veh. Technol.* **68**(1), 856–868 (2018)
7. Yu, Z., Gong, Y., Gong, S., et al.: Joint task offloading and resource allocation in UAV-enabled mobile edge computing. *IEEE Internet Things J.* **7**(4), 3147–3159 (2020)
8. Deng, Y., Chen, Z., Chen, X., et al.: Throughput maximization for multiedge multiuser edge computing systems. *IEEE Internet Things J.* **9**(1), 68–79 (2021)
9. Ale, L., Zhang, N., Fang, X., et al.: Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. *IEEE Trans. Cogn. Commun. Network.* **7**(3), 881–892 (2021)
10. Zhou, J., Zhang, X.: Fairness-aware task offloading and resource allocation in cooperative mobile edge computing. *IEEE Internet Things J.* **9**(5), 3812–3824 (2021)
11. Zhao, N., Ye, Z., Pei, Y., et al.: Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing. *IEEE Trans. Wirel. Commun.* **21**(9), 6949–6960 (2022)
12. Chen, J., Chen, S., Wang, Q., et al.: IRAF: a deep reinforcement learning approach for collaborative mobile edge computing IoT networks. *IEEE Internet Things J.* **6**(4), 7011–7024 (2019)
13. He, X., Wang, S.: Peer offloading in mobile-edge computing with worst case response time guarantees. *IEEE Internet Things J.* **8**(4), 2722–2735 (2020)
14. Wang, Y., Tao, X., Zhang, X., et al.: Cooperative task offloading in three-tier mobile computing networks: an ADMM framework. *IEEE Trans. Veh. Technol.* **68**(3), 2763–2776 (2019)
15. Xu, S., Liu, Q., Gong, B., et al.: RJCC: reinforcement learning based joint communicational and computational resource allocation mechanism for smart city IoT. *IEEE Internet Things J.* **7**(9), 8059–8076 (2020)

16. Zhao, J., Li, Q., Gong, Y., et al.: Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. *IEEE Trans. Veh. Technol.* **68**(8), 7944–7956 (2019)
17. Ho, T.M., Nguyen, K.K.: Joint server selection, cooperative offloading and handover in multi-access edge computing wireless network: a deep reinforcement learning approach. *IEEE Trans. Mob. Comput.* **21**(7), 2421–2435 (2020)
18. Ning, Z., Dong, P., Kong, X., et al.: A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things. *IEEE Internet Things J.* **6**(3), 4804–4814 (2018)
19. Huang, X., Leng, S., Maharjan, S., et al.: Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks. *IEEE Trans. Veh. Technol.* **70**(9), 9282–9293 (2021)
20. Li, H., Xu, H., Zhou, C., et al.: Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment. *IEEE Trans. Veh. Technol.* **69**(9), 10214–10226 (2020)
21. Xu, X., Li, Y., Huang, T., et al.: An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks. *J. Netw. Comput. Appl.* **133**, 75–85 (2019)
22. Li, J., Gao, H., Lv, T., et al.: Deep reinforcement learning based computation offloading and resource allocation for MEC. In: *IEEE Wireless Communications and Networking Conference* (2018)
23. Lyu, X., Tian, H., Sengul, C., et al.: Multiuser joint task offloading and resource optimization in proximate clouds. *IEEE Trans. Veh. Technol.* **66**(4), 3435–3447 (2016)
24. Kozuno, T., Ménard, P., Munos, R., et al.: Learning in two-player zero-sum partially observable Markov games with perfect recall. In: *Advances in Neural Information Processing Systems* (2021)
25. Rashid, T., Samvelyan, M., Schroeder, C., et al.: QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In: *International Conference on Machine Learning* (2018)
26. Lowe, R., Wu, Y., Tamar, A., et al.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: *Advances in Neural Information Processing Systems* (2017)
27. Fu, H., Tang, H., Hao, J., et al.: Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. In: *International Joint Conference on Artificial Intelligence* (2019)



Clustering Based Collaborative Learning Grouping for Knowledge Building

Jiaqi Hao^{1,5}, Weipo Yi^{1,5}, Meirui Ren^{1,3,4,5(✉)}, Chunyu Ai^{2(✉)}, Tianlong Qi^{1,5}, and Longjiang Guo^{1,3,4,5}

¹ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
meiruiren@snnu.edu.cn

² Division of Math and Computer Science, University of South Carolina Upstate, Spartanburg, SC 29303, USA
Aic@uscupstate.edu

³ Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Xi'an 710119, China

⁴ Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China

⁵ Engineering Laboratory of Teaching Information Technology of Shaanxi Province, Xi'an 710119, China

Abstract. With the development of online learning, collaborative learning grouping has been paid more and more attention. In collaborative learning grouping, it is very important to develop new knowledge structures by building knowledge among collaborative learning partners, that is also the key process of knowledge building. For knowledge building in collaborative learning, this paper proposes a clustering grouping method called LGKB, which considers four factors: group size, topic willingness, learning time habit, and cognitive state. Compared with the existing grouping methods, LGKB increased the topic satisfaction and the study time similarity of members within the same group, and it enriched the knowledge structure of different group members. It results in the formation of collaborative learning groups in which students have higher interest and motivation to learn, and it is conducive to knowledge construction and the smooth progress of the collaborative learning process.

Keywords: collaborative learning · knowledge building · study group · hierarchical clustering

1 Introduction

With the outbreak of COVID-19, online learning has become a common choice for most people [11]. For learners on online learning platforms, collaborative learning grouping is an important part of online learning, and learners can

J. Hao and W. Yi—Contributed equally to this work.

carry out various cooperative learning activities without the limitation of time and space [12]. They can not only fully exercise the knowledge they have mastered, but also learn unskilled knowledge from team members [18]. This process is called computer-supported collaborative learning (CSCL) [4]. The process of CSCL needs to promote individual learning and effective collaboration. Learners can consolidate and improve their learning skills by expressing and discussing ideas with each other. It is also the key process of knowledge building [8], which aims to develop new knowledge formations among collaborative learning partners [9].

Obviously, in the process of knowledge building, the knowledge formation of group members and the efficiency of communication among group members have a great impact on the efficiency of collaborative learning. Firstly, complete knowledge formations are helpful for learners to learn diversified knowledge [3]. Therefore, the concepts mastered by members of the same collaborative learning group should be as different as possible to cover more concepts. Secondly, the size of the collaborative learning group and the topic willingness of the group members are the key factors affecting the communication efficiency among the group members: the same topic willingness is conducive to the group members achieving a common learning objective and reducing the disagreement within the group [13]. If the group size is too large, a large number of group members will lead to higher communication costs [19], and the learning efficiency of group members will be reduced [10]. If the group size is too small, a single group member needs to undertake more collaborative learning tasks, which is not conducive to the completion of objectives. Therefore, the size of collaborative learning groups should be controlled in a reasonable range. Finally, learners in CSCL are not limited by time and space [4], so assigning learners with the same learning time habit to the group can also help to improve communication efficiency among group members. To sum up, the formation of collaborative learning groups should take into account group size, topic willingness, learning time habit, and cognitive state.

The existing collaborative learning grouping methods are relatively limited. In these methods, only the test score is usually used as the standard to distinguish the knowledge level, and it is not enough to reflect the learners' real knowledge state, and it is not conducive to knowledge building [19]. Therefore, aiming at knowledge building in CSCL, this paper proposes the LGKB method that comprehensively considers cognitive state, group size, topic willingness, and learning time habit. While ensuring the group size, the LGKB method makes concepts mastered by the group members cover concepts as much as possible. The learning time habits and topic willingness of the group members are as similar as possible.

2 Related Work

In this section, related works on grouping are reviewed. The existing grouping methods can be roughly divided into two types: grouping based on selecting subsets and grouping based on dividing.

2.1 Grouping Based on Selecting Subsets

Grouping based on selecting subsets has one of the most important application scenarios which is the competition scenario. For example, a teacher selects 10 learners from 1000 candidates to form a group to complete a competition together. Selecting subsets requires that x individuals be selected from the set of n individuals to form a team to complete a task together, where x is generally far less than n .

The grouping of competition mainly considers the candidate's skills, costs and other factors of the candidates, and uses the method of a heuristic algorithm. Yadav et al. [20] used the best-fit algorithm that considered skills and costs, to find a group of candidates who are mutually exclusive in skills so that they could complete the task at the lowest cost. Kader et al. [11] applied the Jaya algorithm (a heuristic algorithm) to grouping works. Giorgio Barnabo et al. [4] used a greedy algorithm for grouping which considered group size and skills, and it aimed to find teams with all the skills, and the teams needed to complete the given task. Machado et al. [12] used brute-force algorithm and heuristic algorithm, which considered skills. In the grouping competition scenario, the main factors to be considered are the skills of the participants and the communication costs between the participants. Wen et al. [18] used a neural network to judge whether it was a transactional discussion. Qu et al. [14] used a particle swarm optimization algorithm which took into account the theoretical ability and practical ability of participants, aiming to build the competition team with the highest utility.

Although the grouping strategy based on selecting subsets can select the right members from the candidate set to form an optimal group, only a small number of candidates are included in the group. In teaching scenarios, for the sake of fairness, it is often required that all learners should be included in different groups. We can use the aforementioned selecting subsets algorithms repeatedly to select subsets from the remaining learners, but the groups formed would still be unfair. Therefore, the grouping strategy based on selecting subsets is not suitable for teaching scenarios.

2.2 Grouping Based on Dividing

Grouping based on dividing is mainly used in traditional offline classrooms or online learning platforms, and it requires all learners to be included in different groups. This grouping strategy requires n learners to be divided into m disjoint groups, and the union of m groups is equal to the whole set of n learners.

In traditional classrooms, the number of students in a single class is usually less than 100, and they have the same learning schedule. The main factors considered in grouping are learners' gender, personality, grade points, and skills. Dzvonyar et al. [6] provided a set of grouping standards, which considered many factors such as group size, topic willingness, and learning time habit, and it required each group to have a female at least. Ergin et al. [8] built groups based on homogeneous interests and different skill levels, and ensured that each group

had a highly skilled student at least. Kim et al. [9] only considered one personal characteristics such as credit, age, personality, and so on, and they proposed a mathematical model to form a balanced group to the maximum extent. Unfortunately, the aforementioned methods are all manually grouped, they do not apply to online learning platforms since there are thousands of learners in a single online course.

Andrejczuk et al. [3] considered gender, personality, ability, and group size. A linear programming algorithm was used when the number of learners was small, and a real-time heuristic algorithm was used when the number was large. Nand et al. [13] aimed at forming a group with balanced skills, they considered learners' skill preference and skill level, and they used the firefly algorithm to group. Flores-Parra et al. [10] considered the role type of Belbin and used a method of social networks to group. Xiao et al. [19] used a chat robot to group, and Zhou et al. [23] used a multi-arm slot machine algorithm to group. Rakesh Agrawal et al. [1] considered three factors leadership, achievement, and group size. Chniter et al. [5] combined with ant colony clustering algorithm based on learners' performance for grouping. Sanz-Martínez et al. [17] quantified learners' participation based on webpage views, homework submission, and other activities, and then they used the K-Means algorithm to group learners. O. R. Sánchez et al. [16] proposed a grouping method based on the psychological characteristics of learners in combination with a genetic algorithm. I. M. M. Ramos et al. [15] applied the K-Means algorithm to divide learners with similar learning paths into the same collaborative learning group. I. U. Haq et al. [7] proposed a dynamic grouping method based on learners' learning style and knowledge level. However, the knowledge level here was expressed by learners' test scores and could not reflect learners' mastery of each concept. Y. Zheng et al. [22] proposed a grouping method for different grouping requirements under different educational backgrounds based on an improved genetic algorithm. Akbar et al. [2] proposed a grouping method for improving the formation of student teams (IFST for short) combined with the clustering algorithm to group, they considered the group size and the topic willingness, which achieved good performance. Compared with manual grouping, the above methods are more efficient. However, the four most important factors of cognitive state, group size, topic willingness, and learning time habit are not considered at the same time, and the performance of the formed groups is limited, which is not conducive to knowledge building.

2.3 Summary

For different scenarios, there are so many different grouping methods. However, for a large number of learners, there is a lack of grouping methods that simultaneously consider the four most important factors, and the performance of the formed groups is limited. These grouping methods usually only take the test score as the standard of distinction, which is not enough to reflect the actual knowledge level of learners, and it is not conducive to knowledge building. Given

the above shortcomings, this paper proposes a grouping method that comprehensively considers cognitive state, group size, topic willingness, and learning time habit. It can reflect the real knowledge level of learners and enable the formed collaborative learning groups to perform better. The grouping method is based on the scale-constrained, bottom-up hierarchical clustering algorithm, which has high stability when dealing with massive data and is better able to optimize the grouping effect, so it is suitable for practical collaborative learning grouping scenarios.

This paper proposes the LGKB method and compares LGKB with IFST and RGA, And the experimental results show that LGKB performs best in satisfied degree, time similarity degree, and concept coverage ratio.

3 Problem Formulation

In this section, necessary definitions and formal descriptions of collaborative learning groups are given.

Let $S = \{s_1, s_2, \dots, s_i, \dots, s_N\}$ denote the set of N learners. Set S is divided into M groups $\{G_1, G_2, \dots, G_j, \dots, G_M\}$, here $G_i \cap G_j = \emptyset (i \neq j)$, and $\bigcup_{j=1}^M G_j = S$. Let $P = \{p_1, p_2, \dots, p_l, \dots, p_L\}$ denotes the set of L topics for learners to choose, and $C = \{c_1, c_2, \dots, c_k, \dots, c_K\}$ denotes the set of K concepts. The topic willingness of s_i is denoted as vector $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{il}, \dots, w_{iL}]$, where w_{il} is an integer, represents s_i 's order for topic p_l , this is, s_i selects p_l as the w_{il} -th topic willingness. For example, if there are three topics, i.e. $L = 3$, and s_5 's willingness is $\mathbf{w}_5 = [2, 3, 1]$, it means that s_5 takes p_1 as the second willingness, p_2 as the third willingness, and p_3 as the first willingness. s_i 's learning time habit is denoted as $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{iq}, \dots, t_{iQ}]$, here, the learning time cycle (usually 1 day) is divided into Q time periods, t_{iq} represents the probability that s_i can participate in collaborative learning in the q -th time period, and $\sum_{q=1}^Q t_{iq} = 1$. The cognitive state of s_i is denoted as $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{ik}, \dots, a_{iK}]$, where a_{ik} represents s_i 's proficiency on concept c_k . Finally, let ϵ denotes the upper limit of each group size, δ denotes the lower limit of each group size, here $2 \times (\delta - 1) \leq \epsilon < 2 \times \delta (\delta < \epsilon)$ when setting the upper and lower limits, such as, $\delta = 3$, $\epsilon = 5$, Such a group is called a legitimate group.

Theorem 1. Satisfied Degree: *The proportion of satisfied learners to the total number of learners is called Satisfied Degree (SFD). Students are more satisfied with the topics they are assigned, which contributes to increased interest and motivation in learning. If the group size is between δ and ϵ (including δ and ϵ), and the topic assigned to the group is the learner's first willingness, the learner is considered to be satisfied.*

Theorem 2. Time Similarity Degree: *Time Similarity Degree (TSD) measures the similarity of learning time habits of members in each group. The more similar the study time habits of members within the same group, the more conducive it is for students to engage in collaborative learning anytime, anywhere. TSD is calculated as follows:*

$$TSD = \frac{1}{M} \sum_{j=1}^M \frac{1}{\binom{|G_j|}{2}} \sum_{\substack{s_i, s_v \in G_j \\ i \neq v}} \frac{t_i \bullet t_v}{\|t_i\|_2 \times \|t_v\|_2}, \quad (1)$$

where $\binom{|G_j|}{2}$ denotes combination number of two learners randomly selected from group G_j , “ \bullet ” is the inner product operator of two vectors, “ $\|\cdot\|_2$ ” means to calculate the two-norm of the vector, that is, to sum the squares of each element in the vector and then take the square root.

Theorem 3. Concept Coverage Ratio: Concept Coverage Ratio (CCR) represents the proportion of concepts mastered by members of each group to all concepts. A more comprehensive combination of concepts not only helps the group members learn more concepts but also benefits in achieving the learning goals. It is calculated as follows:

$$CCR = \frac{1}{M} \sum_{j=1}^M \frac{1}{K} \sum_{k=1}^K f(a_k^{(G_j)}, \lambda) \quad (2)$$

where $a_k^{(G_j)}$ denotes the maximum proficiency of members of G_j on c_k . $f(x, \lambda) = 0$ if $x < \lambda$, otherwise $f(x, \lambda) = 1$, here λ is the threshold of concept proficiency. The threshold can be adjusted according to actual needs, if $a_{ik} \geq \lambda$, it is considered that s_i has mastered c_k .

Theorem 4. Problem description: Given the topics set P , the learners set S , the topic willingness vectors set $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_N\}$, the learning time habit vectors set $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N\}$, the cognitive states set $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N\}$, and the lower and upper limits of the group size δ and ϵ . Finally, the learners set S is divided into disjoint collaborative learning groups $\{G_1, G_2, \dots, G_M\}$, which maximizes satisfied degree, time similarity degree and concept coverage ratio.

The important symbols used in this study are summarized in Table 1. Generally, the set is denoted by non-bold italic capital letter, such as S ; the number is denoted by cursive capital letter, such as N ; the vector is denoted by bold lowercase letter, such as \mathbf{w}_i ; the elements in set or vector are denoted by non-bold italic lowercase letters, such as w_{il} .

4 Collaborative Learning Grouping for Knowledge Building

This section mainly introduces the grouping process based on the scale-constrained and bottom-up hierarchical clustering algorithm called LGKB, which mainly includes three steps: the first step, LGKB uses the hierarchical clustering algorithm to group; the second step, it assigns the remaining learners; the third step, it assigns topics to the candidate group to making them become a collaborative learning group.

Table 1. The important symbols used in this paper

Symbol	Description
S	Learners set, $ S = \mathcal{N}$, s_i is the i -th learner
G_j	j -th collaborative learning group
\mathcal{M}	The number of collaborative learning groups
P	Topics set, $ P = \mathcal{L}$, p_l is the l -th topic
C	Knowledge concepts set, $ C = \mathcal{K}$, c_k is k -th concept
\mathbf{w}_i	Topic willingness of s_i , element w_{il} is integer, denotes s_i 's order for topic p_l
\mathbf{t}_i	Learning time habit of s_i , element $0 \leq t_{iq} \leq 1$ denotes the probability that s_i can participate in collaborative learning in the q -th time period
\mathbf{a}_i	Cognitive state of s_i , element $0 \leq a_{ik} \leq 1$ denotes s_i 's proficiency on concept c_k
δ	Group size lower limit
ϵ	Group size upper limit
λ	The threshold of concept proficiency

4.1 Grouping by Hierarchical Clustering Algorithm

The steps of using the clustering algorithm to group are as follows:

Step 1: treat each learner as a separate cluster, with a total of \mathcal{N} clusters.

Step 2: calculate the distance between each two clusters according to the characteristics of clusters.

Step 3: find the two closest clusters, there will be three cases:

If the sum of the number of learners in the two closest clusters is less than ϵ , remove the two clusters from the cluster set and merge them into one cluster, and then the merged new cluster will be added to the cluster set.

If the sum of the number of learners in the two closest clusters is equal to ϵ , then remove the two clusters from the cluster set and merge them into one cluster, then the merged new cluster will be added to the candidate group set as a candidate group.

If the sum of the number of learners in the two closest clusters is greater than ϵ , because the upper and lower bounds satisfy $2 \times (\delta - 1) \leq \epsilon$, there will be at least one cluster in the two closest clusters has learners between δ and ϵ (including δ and ϵ), so the clusters with the number of learners between δ and ϵ are taken as a candidate group and added to the candidate group set, and the clusters with the number of learners not between δ and ϵ are still put back into the cluster set.

Step 4: determine whether the number of clusters in the cluster set is less than δ , if it is greater than or equal to δ , return to step 2, if it is less than δ , stop the loop and process the remaining learners.

In the above steps, whenever a new cluster is generated, the characteristics of the cluster need to be recalculated according to the characteristics of learners in the cluster. The characteristics of a cluster consist of three parts: topic willingness, learning time habit, and cognitive state. The topic willingness of a cluster is the average of the topic willingness of learners in the cluster, and the learning time habit of a cluster is the average of the learning time habit of learners in the cluster. For example, the topic willingness \mathbf{w}_U and the learning time habit \mathbf{t}_U of cluster U are calculated as follows:

$$w_{Ul} = \frac{1}{|U|} \sum_{s_i \in U} w_{il}, \quad t_{Uq} = \frac{1}{|U|} \sum_{s_i \in U} t_{iq}, \quad (3)$$

where w_{Ul} is the element in \mathbf{w}_U , which denotes the average order of U for p_l , t_{Uq} is the element in \mathbf{t}_U , which denotes the average probability of U can participate in collaborative learning in q -th time period.

Since learners can learn concepts that they are not proficient in the same group members. Theoretically, after some period of collaborative learning, the group members can achieve maximum proficiency. Therefore, the maximum proficiency of learners in a certain concept in the cluster is regarded as the proficiency of the cluster in that concept. The cognitive state of cluster U is denoted as \mathbf{a}_U , which is calculated as follows:

$$a_{Uk} = \text{MAX}\{a_{ik} | s_i \in U\} \quad (4)$$

where a_{Uk} is the element in \mathbf{a}_U , which denotes U 's proficiency on c_k .

In step 2, the distance between different clusters U and V is denoted as d_{UV} , which is calculated as follows:

$$d_{UV} = \alpha d_{UV}^{(w)} + \beta d_{UV}^{(t)} + \gamma d_{UV}^{(a)}, \quad (5)$$

where $d_{UV}^{(w)}$, $d_{UV}^{(t)}$ and $d_{UV}^{(a)}$ respectively denote the distance between the two clusters in topic willingness, learning time habit and cognitive state, and α , β , and γ are all non-negative and represent the weights of these three distances respectively, which are specified by the user and have $\alpha + \beta + \gamma = 1$ holds. The three distances are calculated as follows:

$$d_{UV}^{(w)} = \sqrt{\frac{1}{\mathcal{L}} \sum_{l=1}^{\mathcal{L}} \left(\frac{1}{w_{Ul}} - \frac{1}{w_{Vl}} \right)^2}, \quad (6)$$

$$d_{UV}^{(t)} = 1 - \frac{\mathbf{t}_U \bullet \mathbf{t}_V}{\|\mathbf{t}_U\|_2 \times \|\mathbf{t}_V\|_2}, \quad (7)$$

$$d_{UV}^{(a)} = 1 - \sqrt{\frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} (a_{Uk} - a_{Vk})^2}, \quad (8)$$

in Eq. 6, the reciprocal of topic willingness represents the benefit of the topic to learners in the cluster, and $d_{UV}^{(w)}$ is the distance between the two clusters' top

willingness benefits. According to Eq. 6 to Eq. 8, when clustering according to the distance given by Eq. 5, it is easier to assign learners with the same topic willingness and learning time habit and complementary cognitive states to the same group.

In step 3, only the clusters with the number of learners between δ are ϵ are considered as candidate groups, thus ensuring a balanced group size after clustering. In step 4, to prevent the infinite loop, this paper sets the loop termination condition as the number of clusters in the cluster set is less than δ .

4.2 Process the Remaining Learners After Clustering

After clustering, there are at most $\delta \times \epsilon$ learners in the clusters set who are not included in the candidate group. If the clusters set is not empty, there will be two cases:

If the number of learners in the cluster is between δ and ϵ , the cluster is regarded as a candidate group and added to the candidate group set.

If the number of learners in the cluster is less than δ , the learners in this cluster will be reassigned to the candidate group in turn. During reassignment, only the candidate group whose number of learners is greater than or equal to δ and less than ϵ is regarded as the target group, and unassigned learners are added to the target group with the smallest distance from themselves.

After the above steps, all learners will be included in the candidate group.

4.3 Assign Topics to Candidate Groups

The candidate groups are traversed, and the topic with the largest benefit of the group's top willingness is assigned to the group, so that it becomes a collaborative learning group. Denote the willingness benefit of the candidate group U as \mathbf{b}_U . It is calculated as follows:

$$b_{Ul} = \frac{1}{|U|} \sum_{s_i \in U} \frac{1}{w_{il}}, \quad (9)$$

$$b_U = \text{MAX}\{b_{Ul} | p_l \in P\} \quad (10)$$

Here b_{Ul} is the element in \mathbf{b}_U , which represents the willingness gain of the candidate group U on the topic p_l . Clearly, the smaller the ordering of p_l by the learners in the group U , the larger is the willingness gain of p_l to the learners in U .

5 Experiments

5.1 DataSets

Three public real datasets are used in this paper, KDDCup ASSIST2017, and MatMat. The following is an introduction to the datasets.

KDDCup¹: It is a dataset used in the KDD Cup 2010 Educational Data Mining Challenge. It contains records of interactions between learners and computer-aided-tutoring systems during the 2005–2006 academic year.

ASSIST2017²: It is an open dataset collected by ASSISTments online tutoring system, which contains learners' answer records from 2004 to 2007, including learners' time to answer the exercises, their scores on the exercises and other information.

MatMat³: It is collected by adaptive learning group at Faculty of informatics, Masaryk university, it contains the records of learners' answering math problems in the adaptive learning system, the answering time, and the knowledge points contained in the exercises.

5.2 Data Preprocessing

For the above datasets, only the learners who answer more than 15 exercises and the responses of the learners when they answer for the first time are retained, and then the answer records are input into QRCDM [21] to obtain the cognitive state of the learners. The study time pattern of learners is quantified according to the time when they take to solve the exercises. Specifically, first of all, the day is divided into before dawn (00:00–05:00), dawn (05:00–07:00), morning (07:00–11:30), noon (11:30–13:30), afternoon (13:30–18:30), evening (18:30–20:30), and night (20:30–24:00). Then, the number of times each learner answer the exercises in each time period is counted. Finally, The number of answers in each time period is divided by the total number of answers of the learner was normalized, and the normalized vector is taken as the learning time rule of the learner. Due to the lack of relevant information of topic willingness in datasets, the learner's topic willingness is generated by computer simulation. Three simulation methods are uniform distribution, normal distribution, and power law distribution.

5.3 Experimental Configuration

The collaborative learning grouping method proposed in this paper is implemented using Python 3.10.6, Pytorch 1.13.1. The method is implemented based on Intel(R) Core(TM) i5-7200U, with CPU 2.50 GHz and the operating system type of 64 bits.

5.4 Baselines and Metrics

This paper considers the influence of four factors: group size, topic willingness, learning time habit, and cognitive state. RGA and IFST method are selected for comparison. *SFD*, *TSD* and *CCR* are used as evaluation metrics.

¹ <http://pslcdatahop.web.cmu.edu/KDDCup/downloads.jsp>.

² <https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017>.

³ <https://www.fi.muni.cz/adaptivelearning/data/matmat/>.

RGA: Students are randomly divided into several groups between the upper and lower limits of the group size without considering their willingness to choose the topic.

IFST: Hierarchical k-means clustering and weighting formula are used to mine student preferences on topics and make group members reach the maximum allowed group size. Then they are grouped and assigned to topics that match their common willingness.

5.5 Experimental Results and Analysis

In this section, the MatMat dataset is selected to analyze the sensitivity of five hyperparameters: the upper and lower limits of group size, the number of topics, the distribution of topic willingness, the threshold of concept proficiency, and the number of learners as shown in Fig. 1, Fig. 2, and Fig. 3.

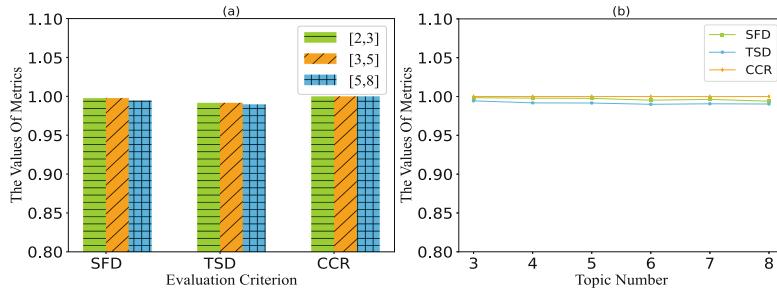


Fig. 1. Figure (a) shows the sensitivity for different group size $[\delta, \epsilon]$, and figure (b) shows the sensitivity for different number of topics

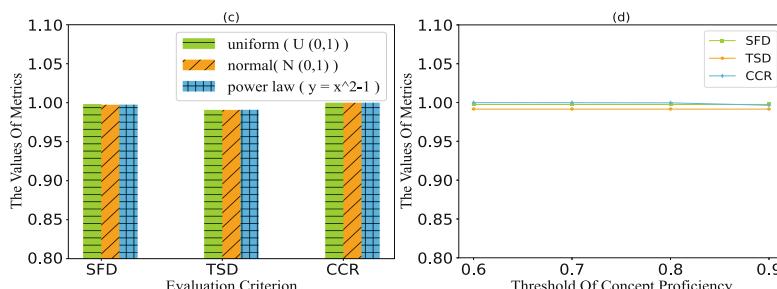


Fig. 2. Figure (c) shows the sensitivity of different distribution methods for topic willingness, figure (d) shows the sensitivity of different the threshold of concept proficiency

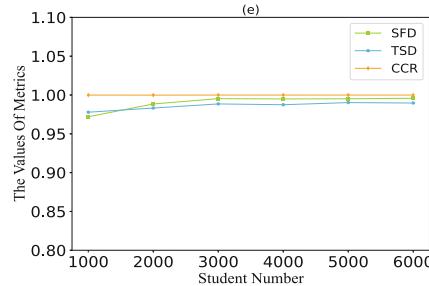


Fig. 3. Figure (e) shows the sensitivity for different numbers of learners

It can be concluded that LGKB proposed in this paper is not sensitive to different group size, the number of topics, the distribution of topic willingness, the threshold of concept proficiency, and the number of learners. According to the sensitivity analysis, the algorithm has good robustness. In this paper, the group size is set as $\delta = 3$ and $\epsilon = 5$. The number of topics is set as 5, the distribution mode of topic willingness is normal distribution, the threshold of concept proficiency is set as 0.7, and the number of learners depends on the dataset.

Table 2. Comparison between LGKB and baselines

Dataset	KDDCup			ASSIT2017			MatMat		
	SFD	TSD	CCR	SFD	TSD	CCR	SFD	TSD	CCR
IFST	0.854	0.600	0.882	0.961	0.789	0.425	0.984	0.216	0.996
RGA	0.478	0.633	0.899	0.452	0.789	0.409	0.476	0.445	0.954
LGKB	0.965	0.955	0.976	0.998	0.964	0.699	0.804	0.763	0.816

After that, LGKB is compared with RGA and IFST, as shown in Table 2. RGA not consider the willingness to choose topics, students are only divided into groups that meet the size of the group. In addition, students in the same group maybe have very different learning time habit, and similar concepts. Leading to data of RGA on the three evaluation metrics is lower and more volatile. IFST considers two aspects of topic willingness and group size, and it is significantly better than the random grouping method in terms of evaluation metrics. The purpose of LGKB proposed in this paper ensures that the group size is met, and it maximizes the three evaluation metrics of satisfied degree, time similarity degree, and concept coverage ratio. And the experiments show that LGKB performs best.

6 Conclusion

In order to improve the efficiency of learners in collaborative learning group and complete knowledge building. This paper proposes a clustering-based grouping

method for collaborative learning called LGKB. LGKB considers four factors: group size, top willingness, learning time habit, and cognitive state. In this paper, learners are preliminarily grouped by the hierarchical clustering algorithm. Then, the remaining learners are divided into the existing set of candidate groups. Finally, topics are assigned to all candidate groups. Then, this paper compares LGKB with RGA and IFST, The three evaluation metrics are as follows: satisfied degree, time similarity degree, and concept coverage ratio. The effectiveness and advancement of LGKB are fully verified. In future work, this paper plans to further consider other factors affecting collaborative learning to improve the efficiency and quality of grouping in collaborative learning based on knowledge building.

Acknowledgments. This work is partly supported by the National Natural Science Foundation of China under Grant No. 61977044; the Ministry of Education's Cooperative Education Project Grant No. 202102591018.

References

1. Agrawal, R., Golshan, B., Terzi, E.: Forming beneficial teams of students in massive online classes. In: ACM Conference on Learning @ Scale (L@S 2014), pp. 155–156. ACM (2014)
2. Akbar, S., Gehringer, E.F., Hu, Z.: Improving formation of student teams: a clustering approach. In: The 40th International Conference on Software Engineering: Companion Proceedings (ICSE 2018), pp. 147–148. ACM (2018)
3. Andrejczuk, E., Bistaffa, F., Blum, C., Rodríguez-Aguilar, J.A., Sierra, C.: Synergistic team composition: a computational approach to foster diversity in teams. *Knowl. Based Syst.* **182**, 104799–104814 (2019)
4. Barnabò, G., Fazzone, A., Leonardi, S., Schwiegelshohn, C.: Algorithms for fair team formation in online labour marketplaces. In: World Wide Web Conference (WWW 2019), Companion of The 2019, pp. 484–490. ACM (2019)
5. Chniter, M., Abid, A., Kallel, I.: Towards a bio-inspired ACO approach for building collaborative learning teams. In: The 17th International Conference on Information Technology Based Higher Education and Training (ITHET 2018), pp. 1–8. IEEE (2018)
6. Dzvonyar, D., Alperowitz, L., Henze, D., Bruegge, B.: Team composition in software engineering project courses. In: The 2nd International Workshop on Software Engineering Education for Millennials (SEEM@ICSE 2018), pp. 16–23. ACM (2018)
7. Haq, I.U., et al.: Dynamic group formation with intelligent tutor collaborative learning: a novel approach for next generation collaboration. *IEEE Access* **9**, 143406–143422 (2021)
8. Huseyin, E.: Instructor-formed capstone teams based on interest and technical experience: the road to success. *J. Comput. Sci. Coll.* **35**, 37–49 (2019)
9. Jong-hwan, K.: A mathematical model for balanced team formation in capstone design class. *J. Eng. Educ. Res.* **21**, 28–34 (2018)
10. Josue-Miguel, F.P., Manuel, C.P., David, E.R., Ricardo, R.C., Carelia, G.P.: Towards team formation using Belbin role types and a social networks analysis approach. In: Technology and Engineering Management Conference (TEMSCON 2018), pp. 1–6. IEEE (2018)

11. Kader, M.A., Zamli, K.Z.: Adopting Jaya algorithm for team formation problem. In: The 9th International Conference on Software and Computer Applications (ICSCA 2020), pp. 62–66. ACM (2020)
12. Machado, L., Stefanidis, K.: Fair team recommendations for multidisciplinary projects. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI 2019), pp. 293–297. ACM (2019)
13. Nand, R., Sharma, A.: Meta-heuristic approaches to tackle skill based group allocation of students in project based learning courses. In: Congress on Evolutionary Computation (CEC 2019), pp. 1782–1789. IEEE (2019)
14. Qu, D., Wu, S.: A competition-oriented student team building method. In: ACM Turing Celebration Conference - China (ACM TUR-C 2019), pp. 82:1–82:2. ACM (2019)
15. Ramos, I.M.M., Ramos, D.B., Gadelha, B.F., Oliveira, E.H.T.D.: An approach to group formation in collaborative learning using learning paths in learning management systems. *IEEE Trans. Learn. Technol.* **14**, 555–567 (2021)
16. Sánchez, O.R., Ordóñez, C.A.C., Duque, M.Á.R., Pinto, I.I.B.S.: Homogeneous group formation in collaborative learning scenarios: an approach based on personality traits and genetic algorithms. *IEEE Trans. Learn. Technol.* **14**, 486–499 (2021)
17. Sanz-Martínez, L., Er, E., Martínez-Monés, A., Dimitriadis, Y., Bote-Lorenzo, M.L.: Creating collaborative groups in a MOOC: a homogeneous engagement grouping approach. *Behav. Inf. Technol.* **38**, 1107–1121 (2019)
18. Wen, M., Maki, K., Dow, S., Herbsleb, J.D., Rosé, C.P.: Supporting virtual team formation through community-wide deliberation. *Proc. ACM Hum. Comput. Interact.* **1**, 109:1–109:19 (2017)
19. Xiao, Z., Zhou, M.X., Fu, W.: Who should be my teammates: using a conversational agent to understand individuals and help teaming. In: The 24th International Conference on Intelligent User Interfaces (IUI 2019), pp. 437–447. ACM (2019)
20. Yadav, A., Sairam, A.S., Kumar, A.: Concurrent team formation for multiple tasks in crowdsourcing platform. In: Global Communications Conference (GLOBECOM 2017), pp. 1–7. IEEE (2017)
21. Yang, H., et al.: A novel quantitative relationship neural network for explainable cognitive diagnosis model. *Knowl.-Based Syst.* **250**, 109156 (2022)
22. Zheng, Y., Li, C., Liu, S., Lu, W.: An improved genetic approach for composing optimal collaborative learning groups. *Knowl.-Based Syst.* **139**, 214–225 (2018)
23. Zhou, S., Valentine, M., Bernstein, M.S.: In search of the dream team: temporally constrained multi-armed bandits for identifying effective team structures. In: The 2018 CHI Conference on Human Factors in Computing Systems (HFCS 2018), pp. 1–13. ACM (2018)



Unsupervised Feature Selection via Fuzzy K-Means and Sparse Projection

Kun Jiang¹(✉), Lei Zhu¹, and Qindong Sun^{1,2}

¹ School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China

{jk_365,leizhu}@xaut.edu.cn, sqd@xjtu.edu.cn

² School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China

Abstract. Recent years, unsupervised feature selection (UFS) has obtained widespread attention in various tasks of high-dimensional data mining. However, how to characterize the potential structural information of unlabeled data samples remains an unresolved challenging problem. Some existing UFS models explore the manifold structure and hard pseudo-labels in the high-dimensional feature space, which overlook the noisy information and the data fuzziness. In this paper, we propose an unsupervised feature selection model based on fuzzy K-Means and sparse projection (FKMSP). In particular, the model first employs fuzzy K-Means to obtain discriminative pseudo-labels for data samples that considers the fuzzy distance between data samples and cluster centroids. Then, through an regressive fitness term with $l_{2,p}(0 < p < 1)$ norm constraint for the soft pseudo-labels, the sparse projection matrix is obtained which can effectively represent feature importance. A simple yet efficient iterative optimization algorithm is developed to solve the objective function, together with empirical verified convergence. Extensive experimental results on the benchmark databases demonstrate the effectiveness and superiority of the proposed FKMSP model compared with other state-of-the-art models.

Keywords: Unsupervised feature selection · Fuzzy K-Means · Clustering centroid · Pseudo-label learning

1 Introduction

Unsupervised feature selection (UFS) has attracted increasingly attention by leveraging the structure information to compensate the lack of pseudo label information for large amounts of high-dimensional data [1, 10]. As an efficient dimensionality reduction method, unsupervised feature selection technology can provide interpretable high-quality features [15, 26]. Unsupervised feature selection can be roughly divided into three categories: filtering methods, wrapper methods, and embedded methods [24]. However, in unsupervised feature selection methods, how to mine the structure of data is an effective method to compensate for the lack of label information. Embedded methods deploy the feature

selection procedure into some specific model for all data features, and the important features can be intermediately determined with some ranking metrics when the devised model is finally learned. During the past years, many embedded UFS models have been proposed to explore the underlying data distribution and discriminative information.

The manifold structure is always characterized by the neighborhood graph of data samples, and the graph is constructed by measuring the similarity between samples. Recently, researchers have proposed various methods to generate pseudo labels, including spectral embedding, spectral clustering, matrix factorization, and dictionary learning, to name a few [13, 20]. Recent years, to compensate the lack of label information, pseudo-label learning has become an critical method for UFS models, by learning the cluster membership for unlabeled samples [18, 20]. Essentially, the underlying structure of data can be effectively characterized by assigning each sample to its closely cluster [15]. Nonetheless, the traditional pseudo-label with hard labels learning to describe the crisp cluster is not consistent with the natural of data distribution. The learned hard label learning ignores the fact that sample data may belong to multiple clusters, resulting in significant information loss and a deteriorated performance in feature selection. Besides that, the results of manifold structure mining and pseudo label learning directly conducted in high-dimensional space may deviate significantly due to the influence of noise and redundant information.

In view of these issues, we propose an effective unsupervised feature selection model based on fuzzy K-Means and sparse projection (FKMSP). Specifically, the underlying cluster structure is captured by K-means and the integration of data fuzziness [5, 22]. The projection learning method could determine the low-dimensional subspace for data samples with a constrained projection matrix, which has the advantages of characterizing the natural graph structure for multi-subspaces with low-dimensional representation [20]. Then, some nonlinear constraints including sparse, orthogonal or discrete traits on projection matrix could further facilitate the feature selection procedure [13, 19, 20]. In summary, by considering the data fuzziness and projection learning, the FKMSP has an enough capability of the subspace exploration and robust regression for selected features. The contributions of this paper are outlined as follows:

1. We propose a novel UFS model via fuzzy K-Means and $l_{2,0}$ -norm sparse projection, which could capture the data fuzziness and enhance the robustness to noisy features.
2. An efficient alternative optimization algorithm with generalized soft-thresholding (GST) is developed to solve the challenging objective function with satisfactory convergence.
3. Experimental results on benchmark databases demonstrate the effectiveness and superiority of the proposed FKMSP model and the optimization algorithm.

The rest of this paper is organized as follows. We briefly review some related works in Sect. 2. In Sect. 3, we elaborate the details of the proposed model. We

develop an efficient alternating algorithm in Sect. 4. In Sect. 5, we conduct experiments to illustrate the effectiveness and superiority of the proposed method. And Sect. 6 concludes this paper.

2 Related Work

2.1 Unsupervised Feature Selection

Embedded models combines machine learning and unsupervised feature selection (UFS) methods into a single objective function for optimization. For example, MCFS utilizes the multi-cluster manifold structure of data for feature selection [1]. UDFS combines discriminative analysis with $\ell_{2,1}$ regularization to select discriminative features and minimize the ℓ_1 -norm within a joint framework [24]. NDFS explicitly imposes non-negative constraints on class metrics learned through spectral clustering to discover the intrinsic manifold structure for unsupervised feature selection [8]. EUFS incorporates feature selection directly into the clustering process using sparse learning, eliminating the need for transformations [21]. SOGFS performs local structure learning and feature learning for feature selection [15]. RNE is presented to obtain weight indicated features via the ℓ_1 -norm sample reconstruction loss with predefined local linear embedding information [10]. However, it is evident that these UFS methods overlook the fact that data ambiguity is ubiquitous in both nature and human society. This means that data samples may belong to multiple clusters with varying probabilities simultaneously. These issues can result in substantial information loss and irrelevant feature subsets, ultimately leading to suboptimal clustering performance. Nonetheless, there exist two problems for these traditional embedded UFS models. First, these UFS models overlook the data fuzziness w.r.t. different clusters. Its performance often highly depends on the generation quality of the initial label, which would lead to a large amount of discriminative information loss. Second, a variety of models select features in original feature space without the guidance of underlying subspace distribution, which suffers from the noisy and redundant features.

2.2 Fuzzy K-Means Clustering

The fuzzy K-Means model considers the fuzziness of cluster indicator for each data sample, which is realized by calculating the Euclidean distance between data sample and cluster centroid [16, 17]. The difference of various FKM methods lies in the choice of regularization term [9, 22], norm-based distance metric [3, 4, 25] and optimizing solution [5, 23]. In FKM, different regularization terms are utilized to sparse the membership matrix to avoid discrete solutions and retain the most possible membership relations. For instance, Xu et al. propose a robust and sparse fuzzy K-Means (RSFKM) model, in which the Frobenius norm is employed as the regularization term to adjust the sparsity of the membership matrix [22]. Some researchers consider maximum entropy FKM based

on the maximum entropy regularization [9]. To compensate the limitations of l_2 -norm in traditional FKM, different metrics of distance between data samples and clustering centers are leveraged to suppress its sensitivity to outliers and noisy features. Recent advances in FKM is about the optimization algorithm such as initialization and auxiliary points to find better gradient direction [2, 23]. The soft-label matrix can be obtained efficiently by approximating the objective function with fractional programming and solvable surrogate function to achieve lower objective function value [4, 23]. However, the fuzzy membership matrix learned in the original feature space suffer from the noisy and redundant features, which may result in feature importance confusion and over-fitting issues for some embedded tasks [18, 20]. Therefore, it has been proven to be effective to consider projection learning in FKM model to achieve robust and compact subspace distribution (UDPFS) [20]. However, the ideal row-sparsity constraint on projection is too strict to characterize the cluster centroids in subspaces, which brings about complicated parameter tuning problems in feature selection.

3 The Proposed Model

3.1 Unsupervised Embedded Feature Selection

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ denotes data matrix with n samples, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i th sample with d features. The pseudo-label matrix is denoted as $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]^T \in \mathbb{R}^{n \times c}$, where $\mathbf{y}_i \in \mathbb{R}^c$ is the label vector for each data sample and c is the cluster number. For the hard-label constraint, $y_{ij} = 1$ indicates that \mathbf{x}_i belongs to the j th cluster, and $y_{ij} = 0$ otherwise. By leveraging the discriminative label information, the cluster distribution can be exploited to clearly characterize the structure of all data samples. Under various transformation, the discriminative information could be maintained persistently with some extra constraints. To map each data point to its label vector, we present a generalized flexible embedded model for feature selection as follows:

$$\min_{\mathbf{W}, \mathbf{Y} \in \Pi} \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \gamma \mathcal{Q}(W) + \mathcal{R}(\mathbf{Y}) \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{d \times c}$ is the projection matrix, $\mathcal{Q}(W)$ denotes the regularization term of W to avoid trivial solution, and $\mathcal{R}(\mathbf{Y})$ is introduced to balance the probability of the membership matrix \mathbf{Y} that the sample belongs to different clusters.

The regularization $\mathcal{Q}(W)$ for the projection matrix are determined by various tasks, such as Frobenius norm for classification, l_1 -norm for image restoration and $l_{2,0}$ -norm for feature selection [10, 13]. For feature selection task, the row-sparsity on the projection matrix could be employed to reflect features importance. However, it is generally hard to solve $l_{2,0}$ -norm constrained problem due to its nonconvexity. In this paper, we introduce $\mathcal{Q}(W) = \|\mathbf{W}\|_{2,p}^p = \sum_{j=1}^d \|\mathbf{w}^j\|_2^p$ to obtain a sparse flexible embedding for UFS task, where $\mathbf{w}^j \in \mathbb{R}^{1 \times c}$ denotes the j th row of \mathbf{W} . It has been proved that the flexible $p(0 < p \leq 1)$ could balance the row-sparsity and model convexity for better feature subsets [19].

3.2 Fuzzy K-Means for Embedding Learning

As discussed above, traditional UFS methods have serious deficiencies in considering the discrimination of data samples due to the lack of label information. The pseudo-label learning is an effective method to enhance the discrimination for UFS models. Therefore, the $\mathcal{R}(\mathbf{Y})$ in Eq. (1) can be realized by different pseudo-label learning. However, research has proved the limitations of the crisp approaches when performing clustering in the original data space [9, 22]. Actually, these models overlook the data fuzziness w.r.t. different clusters. And its performance often highly depends on the quality of the initial membership matrix, which would lead to a large amount of discriminative information loss.

In this paper, we propose to utilize the fuzzy K-Means clustering to obtain the soft-label embedding matrix for all data samples. Therefore, we implement the regularizer $\mathcal{R}(\mathbf{Y})$ in Eq. (1) as follows:

$$\begin{aligned} \min_{\mathbf{m}_j, y_{ij}} \quad & \sum_{i=1}^n \sum_{j=1}^c \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 y_{ij} + \alpha \|\mathbf{Y}\|_F^2 \\ \text{s.t.} (\forall i) \quad & \mathbf{y}_i^T \mathbf{1} = 1, 0 \leq y_{ij} \leq 1 \end{aligned} \quad (2)$$

where α is the parameter to promote the fuzziness of membership matrix and $\Pi = \{\mathbf{Y} | (\forall i) \mathbf{y}_i^T \mathbf{1} = 1, 0 \leq y_{ij} \leq 1\}$ constrains each row of \mathbf{Y} to have sum-to-one possibilities of being assigned to all clusters. Then, the trivial solution of $\mathbf{W} = \mathbf{O}$ and $\mathbf{Y} = \mathbf{O}$ could be effectively avoided.

3.3 The Objective Function

As stated above, the unsupervised feature selection with sparse flexible embedding framework determine feature subset with robust reconstruction advantage. Then, the fuzzy K-Means clustering facilitates the pseudo-label matrix with interpretable ambiguity of the cluster indicator vector for practical data sample. As a result, the objective function of the proposed FKMSp model is formulated as follows:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{M}, y_{ij}} \quad & \sum_{i=1}^n \sum_{j=1}^c \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 y_{ij} + \alpha \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \gamma \|\mathbf{W}\|_{2,p}^p \\ \text{s.t.} \quad (\forall i) \quad & \mathbf{y}_i^T \mathbf{1} = 1, 0 \leq y_{ij} \leq 1 \end{aligned} \quad (3)$$

The joint minimization of fuzzy K-Means and flexible embedding with $l_{2,p}$ -norm sparse regularization (3) enables the projection W to measure the importance and relevance of features. By enabling soft-label leaning for UFS, the ambiguity cluster structure for all data samples could be exploited to resist noisy features and outliers in the original space. And it is quite reasonable and effective to impose the $l_{2,p}$ -norm constraint on projection matrix W , which could seamlessly regress high-dimensional data samples to the sparse soft-label matrix.

4 Optimization Algorithm

4.1 Optimizing Procedure

Due to the $l_{2,p}$ -norm and the non-smooth constraint, it is not trivial to solve the non-convex and non-smooth objective function in Eq. (3). Therefore, we alternatively update the variables \mathbf{W} , \mathbf{M} and \mathbf{Y} with three smoothed subproblems.

Updating \mathbf{W} by Fixing \mathbf{M} and \mathbf{Y} . With fixed \mathbf{M} and \mathbf{Y} , the problem (3) can be transformed into

$$\min_{\mathbf{W}} \quad \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \tau \|\mathbf{W}\|_{2,p}^p \quad (4)$$

where $\tau = \frac{\gamma}{\alpha}$. The iterative reweighted method has been proved to converge and lead to stationary points after several iteration [11, 12]. However, study shows it will get stuck into local optimal solution which means it relies heavily on the initializations [14, 23]. Inspired by recent advances on solving $l_{2,p}$ -norm problems [26, 27], we propose to update the problem using Augmented Lagrangian Multiplier (ALM) method.

According to ALM method, we first introduce an auxiliary variable \mathbf{G} with the constraint $\mathbf{W} = \mathbf{G}$, and then reformulate the subproblem (4) into its augmented Lagrangian function as follows:

$$\min_{\mathbf{W}, \mathbf{G}} \quad \|\mathbf{X}^T \mathbf{G} - \mathbf{Y}\|_F^2 + \tau \|\mathbf{W}\|_{2,p}^p + \frac{\mu}{2} \|\mathbf{G} - \mathbf{W} + \frac{1}{\mu} \mathbf{A}\|_F^2 \quad (5)$$

where μ is the penalty parameter and \mathbf{A} is the Lagrangian multiplier. Then, we iteratively update the two variable by fixing the other one, which is mainly specified into two steps.

Step 1: Update \mathbf{G} . When fixed \mathbf{W} , (5) becomes

$$\min_{\mathbf{G}} \quad \|\mathbf{X}^T \mathbf{G} - \mathbf{Y}\|_F^2 + \frac{\mu}{2} \|\mathbf{G} - \mathbf{W} + \frac{1}{\mu} \mathbf{A}\|_F^2 \quad (6)$$

By setting the derivative of (6) w.r.t. \mathbf{G} , we can obtain

$$\mathbf{G} = (2\mathbf{X}\mathbf{X}^T + \mu\mathbf{I})^{-1}(2\mathbf{X}\mathbf{Y} + \mu\mathbf{W} - \mathbf{A}) \quad (7)$$

Step 2: Update \mathbf{W} . When fixed \mathbf{G} , (5) becomes

$$\min_{\mathbf{W}} \quad \frac{1}{2} \|\mathbf{W} - \mathbf{Q}\|_F^2 + \frac{\tau}{\mu} \|\mathbf{W}\|_{2,p}^p \quad (8)$$

where $\mathbf{Q} = \mathbf{G} + \frac{1}{\mu} \mathbf{A} \in \mathbb{R}^{d \times c}$. According to the definition of $l_{2,p}$ -norm, problem (8) can be row-wisely separated into d subproblems as follows:

$$\min_{\mathbf{w}^i} \quad \frac{1}{2} \|\mathbf{w}^i - \mathbf{q}^i\|_2^2 + \frac{\tau}{\mu} \|\mathbf{w}^i\|_2^p \quad (9)$$

Let the SVD of row vector \mathbf{w}^i as $\mathbf{w}^i = [1]\sigma(\mathbf{w}^i)\mathbf{v}^T$, where $[1]$ denotes a matrix with one element as 1, and $\sigma(\mathbf{w}^i)$ is the only singular value of \mathbf{w}^i . Then, we have the following deviation, $\|\mathbf{w}^i\|_2 = (\mathbf{w}^i\mathbf{w}^i)^{\frac{1}{2}} = (\sigma(\mathbf{w}^i)\mathbf{v}^T\mathbf{v}\sigma(\mathbf{w}^i))^{\frac{1}{2}} = \sigma(\mathbf{w}^i)$ and $\mathbf{w}^i = [1]\sigma(\mathbf{w}^i)\frac{\mathbf{w}^i}{\|\mathbf{w}^i\|_2}$.

Therefore, problem (9) can be rewritten as

$$\min_{\mathbf{w}^i} \quad \frac{1}{2}\|\mathbf{w}^i - \mathbf{q}^i\|_2^2 + \frac{\tau}{\mu}\sigma(\mathbf{w}^i)^p \quad (10)$$

By leveraging Theorem 1 in [7], the solution of this problem is given by $\mathbf{w}^i = \tilde{\mathbf{u}}\sigma^*\tilde{\mathbf{v}}^T$, where $\tilde{\mathbf{u}} = [1]$ and $\tilde{\mathbf{v}} = \frac{\mathbf{q}^i}{\|\mathbf{q}^i\|_2}$ are the left and right singular vectors of \mathbf{q}^i . And σ^* can be obtained by solving the following nonconvex problem,

$$\min_{\sigma > 0} \quad \frac{1}{2}\|\sigma - \sigma(\mathbf{q}^i)\|_2^2 + f(\sigma) \quad (11)$$

where $f(x) = \frac{\tau}{\mu}x^p (0 < p < 1)$ is a non-decreasing concave function. This problem can be efficiently solved by Difference of Convex (DC) Programming [7] or Generalized Iterated Shrinkage Algorithm (GISA) [27] in Theorem 1.

Theorem 1. *The Generalized Iterated Shrinkage Algorithm (GISA) solves the following nonconvex problem*

$$\begin{aligned} \min_{x \geq 0} \quad & \frac{1}{2}\|x - \sigma\|_2^2 + \lambda x^p \\ \text{s.t.} \quad & 0 < p < 1 \end{aligned} \quad (12)$$

by utilizing the Generalized Soft Thresholding (GST) operator $\tau_p^{GST}(\lambda)$ defined as:

$$\tau_p^{GST}(\lambda) = (2\lambda(1-p))^{\frac{1}{2-p}} + \lambda p(2\lambda(1-p))^{\frac{p-1}{2-p}} \quad (13)$$

Then, the optimal solution of x is

$$\begin{cases} x = 0, & \sigma \leq \tau_p^{GST}(\lambda) \\ x = \text{sign}(\sigma)S_p^{GST}(\sigma, \lambda), & \text{otherwise} \end{cases} \quad (14)$$

where $S_p^{GST}(\sigma, \lambda)$ is obtained by solving $S_p^{GST}(\sigma, \lambda) - \sigma + \lambda p(S_p^{GST})^{p-1} = 0$ with fixed point iteration method in the range of $((2\lambda(1-p))^{\frac{1}{2-p}}, +\infty)$.

In summary, the updating procedure of \mathbf{W} is summarized in Algorithm 1, where the Lagrangian multiplier \mathbf{A} and penalty parameter μ are updated accordingly.

Updating \mathbf{M} by Fixing \mathbf{W} and \mathbf{Y} . With fixed \mathbf{W} and \mathbf{Y} , the problem (3) can be rewritten as follows:

$$\min_{\mathbf{M}} \quad \sum_{i=1}^n \sum_{j=1}^c \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 y_{ij} \quad (15)$$

Algorithm 1. ALM Method for Subproblem (4)**Input:** $\mathbf{X}, \mathbf{Y}, \tau$;1: Initialize $\mu > 0$, \mathbf{A} and $1 < \rho < 2$.2: **while** not converged **do**3: Update \mathbf{G} according to (6);4: Update \mathbf{W} by solving (8);5: Update \mathbf{A} by $\mathbf{A} = \mathbf{A} + \mu(\mathbf{G} - \mathbf{W})$;6: Update μ by $\mu = \min\{\rho\mu, \mu_{max}\}$.7: **end while****Output:** \mathbf{W}

By taking the derivative of Eq. (15) for each \mathbf{m}_j ($1 \leq j \leq c$) and setting it to zero, we have

$$\mathbf{m}_j = \frac{\sum_{i=1}^n y_{ij} \mathbf{x}_i}{\sum_{i=1}^n y_{ij}} \quad (16)$$

Updating \mathbf{Y} by Fixing \mathbf{W} and \mathbf{M} . With fixed \mathbf{W} and \mathbf{M} , the problem (3) can be transformed into

$$\begin{aligned} \min_{y_{ij}} \quad & \sum_{i=1}^n \sum_{j=1}^c \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 y_{ij} + \alpha \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 \\ \text{s.t.} \quad & (\forall i) \quad \mathbf{y}_i^T \mathbf{1} = 1, 0 \leq y_{ij} \leq 1 \end{aligned} \quad (17)$$

Similar to the updating step of \mathbf{M} , for each subproblem (17), we reformulate it into the following vector form,

$$\begin{aligned} \min_{\mathbf{y}_i} \quad & \|\mathbf{y}_i - \mathbf{W}^T \mathbf{x}_i + \frac{\mathbf{d}_i}{2\alpha}\|_2^2 \\ \text{s.t.} \quad & \mathbf{y}_i^T \mathbf{1} = 1, 0 \leq y_{ij} \leq 1 \end{aligned} \quad (18)$$

where $\mathbf{d}_i \in \mathbb{R}^c$ denotes the distance vector between \mathbf{x}_i and all c cluster centroids, i.e., $d_{ij} = \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$. Then, this problem can be solved efficiently via the iterative algorithm proposed in [6].

Based on the updating steps, the whole alterative optimization rules are shown in Algorithm 2.

5 Experiments

In this section, we perform comparison experiments of feature selection and clustering to validate the superiority of our approach.

5.1 Experimental Setup

In the experiments, we utilize eight benchmark datasets, including three human face image datasets (namely, YALE, JAFFE, and ORL), a handwritten digit

Algorithm 2. Alternative Optimization for Problem (3)**Input:** \mathbf{X} cluster number c , α and γ .1: Initialize \mathbf{Y} and \mathbf{W} with random matrix satisfying the constraints.2: **while** not converged **do**3: Fix \mathbf{M} and \mathbf{Y} , update \mathbf{W} by Algorithm 1.4: Fix \mathbf{W} and \mathbf{Y} , update \mathbf{M} by Eq. (16).5: Fix \mathbf{W} and \mathbf{M} , update \mathbf{Y} by solving (18).6: **end while****Output:** Sort $\|\mathbf{w}^i\|_2$ in descending order, and select the top h ranked features**Table 1.** The statistical information of benchmark datasets.

Datasets	#Samples	#Features	#Classes	Type
YALE	165	1024	15	image
JAFFE	213	676	10	image
ORL	400	1024	40	image
USPS	9298	256	10	digit
LUNG	203	3312	5	biological
COIL20	1440	1024	20	object

dataset (USPS), a biology dataset (LUNG), and an object image dataset (COIL20). The statistical information of the datasets are presented in Table 1.

We compare the proposed FKMSP model with eight state-of-the-art UFS approaches and one baseline that use all original features. Among these UFS competitors, there are graph-based models such as MCFS [1], UDFS [24] and SOGFS [15] along with recent robust approaches including RNE [10], USFS [18] and UDPFS [20]. For the graph-based models, the neighborhood size is initialized with $k = 5$. The comparison models are carefully tuned according to the parameter settings specified in the reference. The involved parameters in the FKMSP model is first searched from the set $\{10^{-5}, 10^{-4}, \dots, 10^4, 10^5\}$ to obtain preferable parameters, and then finely tuned within a small range according to preliminary results. The value of p is set to 1 to approximate $l_{2,0}$ row-sparsity. The number of selected features are tuned from the range $[20, 30, \dots, 90, 100]$ for different models.

Finally, we employ K-Means method to cluster the selected features. To evaluate the clustering performance, we adopt two widely used evaluation criteria: clustering accuracy (ACC) and normalized mutual information (NMI). Since the objective function constitutes non-convex term, we run the proposed method 40 times to ensure the stability of the model. Then, we record stable average results with standard deviations.

5.2 Results and Analysis

The clustering results on real-world datasets are shown in Table 2 and Table 3. From the experimental results, we can briefly obtain the following observations.

1. All the feature selection methods are effective for clustering in most cases. As the redundant and noisy features exist in the high-dimensional features, the clustering performance of the All-Fea method will be deteriorated inevitably. It is quite critical to select a few informative features with some excellent UFS methods that leveraging the underlying data structure.
2. Generally speaking, the graph learning methods such as SOGFS have achieved promising clustering results compared to methods that based on fixed graph in the original space such as MCFS and UDFS. The RNE models have also obtained satisfactory results by leveraging the robust l_1 -norm sample-wise reconstruction assumptions. The soft-label learning via fuzzy k-means could facilitates the exploration of underlying subspace distribution by measuring the similarities between each data sample and the clustering centroids.
3. The proposed FKMSP method exceeds other competitors on all datasets, especially the two soft-label learning methods, i.e., USFS and UDPFS. This is mainly due to the fuzzy k-means clustering for soft-label learning and the adjustable $l_{2,p}$ projection for sparse regression. While the row sparse projection is employed to select features in USFS and UDPFS methods, which is too strict to approximate the high-dimensional cluster centroids.

Overall, the experimental findings on real-world data have demonstrated the superiority of the FKMSP method compared to the state-of-the-arts.

Table 2. Clustering results (ACC) \pm std% of different unsupervised feature selection methods on benchmark datasets.

Datasets	YALE	JAFFE	ORL	USPS	LUNG	COIL20
All-Fea	40.91 ± 5.77	78.48 ± 4.66	53.04 ± 2.46	61.35 ± 1.31	72.60 ± 5.68	63.02 ± 3.49
MCFS	39.80 ± 2.65	80.15 ± 6.05	51.28 ± 2.78	62.59 ± 4.55	78.54 ± 5.79	63.32 ± 3.82
UDFS	34.24 ± 1.00	81.28 ± 2.61	46.35 ± 2.19	40.79 ± 1.81	78.83 ± 2.33	62.56 ± 3.91
SOGFS	36.46 ± 1.24	83.90 ± 4.21	52.68 ± 2.57	50.22 ± 3.96	78.03 ± 2.29	63.12 ± 2.18
RNE	38.38 ± 3.37	83.69 ± 4.73	52.00 ± 2.19	62.73 ± 2.51	78.45 ± 6.58	62.54 ± 3.14
USFS	47.81 ± 2.34	80.72 ± 2.17	55.77 ± 3.03	63.18 ± 2.12	81.28 ± 3.07	64.18 ± 3.45
UDPFS	47.69 ± 2.98	81.93 ± 3.55	54.21 ± 2.91	62.37 ± 2.05	79.67 ± 3.08	61.77 ± 1.99
FKMSP	52.12 ± 3.65	84.87 ± 5.07	58.00 ± 3.55	64.30 ± 2.48	82.65 ± 3.22	66.19 ± 2.85

5.3 Parameter Sensitivity and Convergence Validation

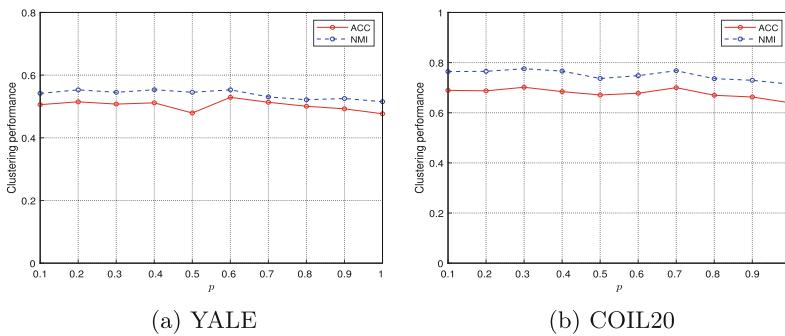
There are two regularization parameters α and γ in the FKMSP model, where α controls the soft-label regression and γ balances the sparsity level of the projection matrix. We finely tuned the parameters combination with grid search

Table 3. Clustering results (NMI) \pm std% of different unsupervised feature selection methods on benchmark datasets.

Datasets	YALE	JAFFE	ORL	USPS	LUNG	COIL20
All-Fea	46.95 ± 4.13	83.86 ± 3.26	74.18 ± 1.55	61.40 ± 1.25	68.53 ± 4.10	76.30 ± 1.98
MCFS	45.56 ± 1.60	81.71 ± 3.40	71.49 ± 1.92	58.00 ± 2.28	70.83 ± 5.51	75.64 ± 1.60
UDFS	40.35 ± 1.64	82.77 ± 2.62	70.04 ± 1.31	34.85 ± 2.93	71.55 ± 4.76	73.93 ± 3.01
SOGFS	43.48 ± 1.50	85.62 ± 2.19	71.95 ± 1.49	53.98 ± 3.77	71.17 ± 2.90	73.55 ± 3.02
RNE	47.44 ± 2.63	84.68 ± 6.03	70.54 ± 2.64	61.79 ± 2.27	68.24 ± 4.37	69.44 ± 4.78
USFS	52.37 ± 2.59	83.11 ± 4.16	72.52 ± 2.39	60.81 ± 3.14	66.52 ± 2.88	71.33 ± 2.71
UDPFS	54.91 ± 2.54	82.77 ± 4.21	73.19 ± 3.08	59.94 ± 3.02	67.38 ± 4.96	72.67 ± 4.02
FKMSP	55.68 ± 1.75	85.73 ± 2.88	74.64 ± 2.85	62.27 ± 2.51	71.05 ± 2.46	74.32 ± 3.51

strategy and plot the clustering ACCs with 3D bar by varying α and γ in Fig. 2. The number of selected features is set to 60. As can be observed, the performance of FKMSP model maintains consistent variation across the JAFFE, USPS, LUNG and COIL20 datasets, while exhibiting some fluctuations on the remaining datasets. Specifically, the clustering results are comparatively sensitive to γ , which is due to the complicated sparsity effect of the projection matrix. While α has played a stable role on the performance of the proposed model which verify the critical importance of the soft-label learning.

As the $l_{2,p}$ -norm ($0 < p \leq 1$) constraint has contributed a lot to the approximation of the $l_{2,0}$ row-sparsity, along with the parameter γ that controls the importance of the sparse regularization term. To investigate the sensitivity of hyper-parameter p , we carry out feature selection experiments on YALE and COIL20 datasets by varying hyper-parameter p within the range $[0.1, 0.2, \dots, 0.9, 1]$, where the rest parameters are set to the optimal empirical values. For each value of p , we repeat the experiment for 20 times and report the average ACC and NMI in Fig. 1. Generally speaking, good performance can

**Fig. 1.** The clustering performance of the FKMSP model with different values of p .

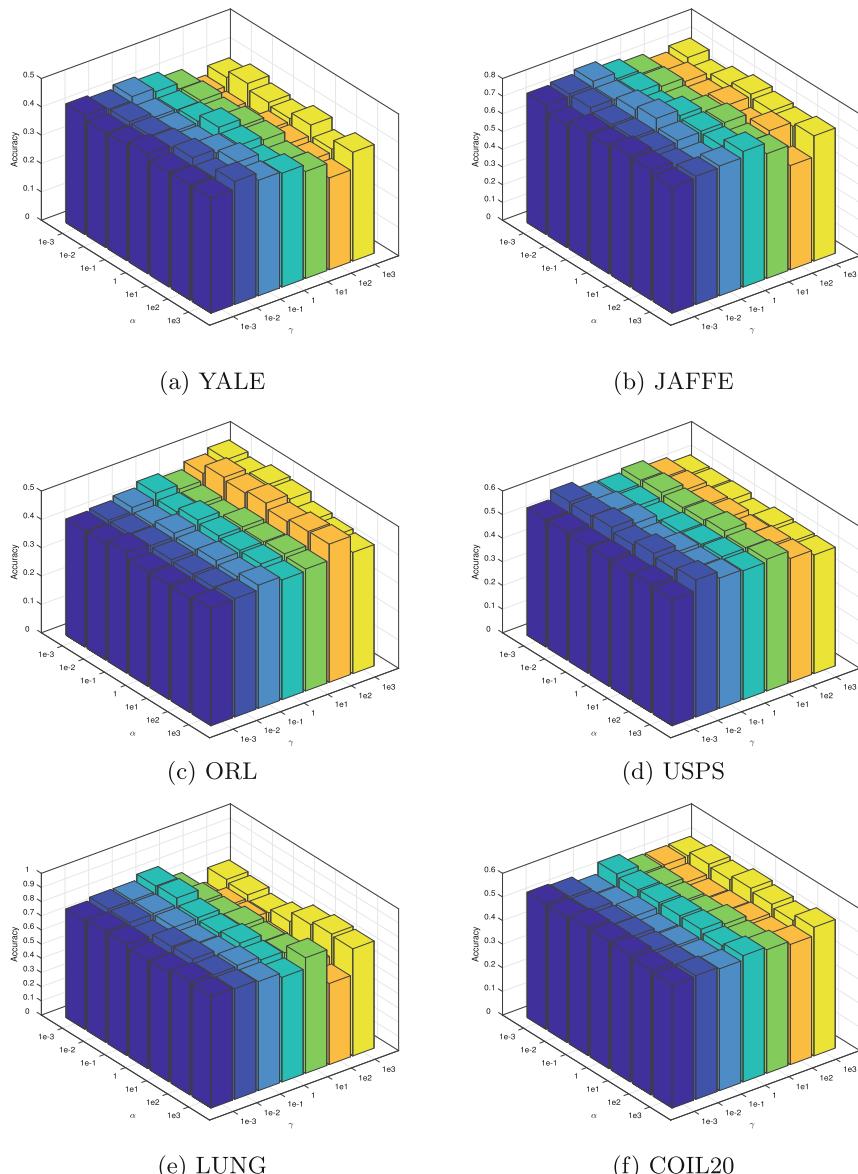
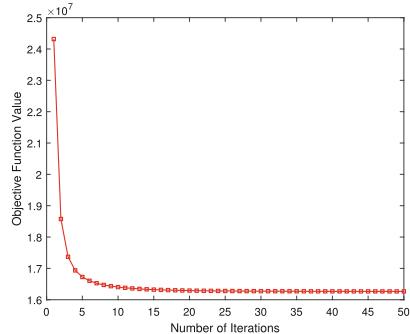
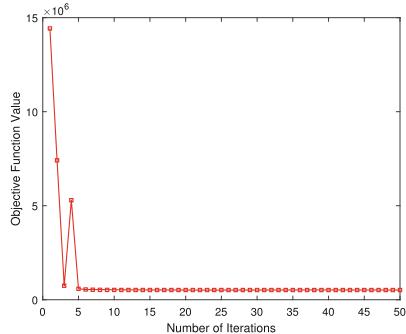


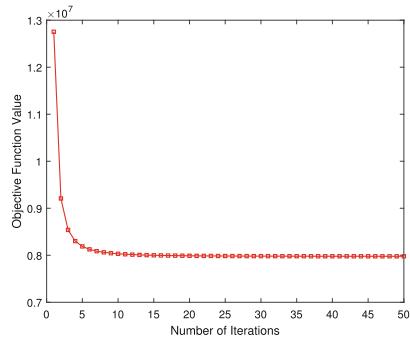
Fig. 2. The ACC variation of FKMSP model with different α and γ on six benchmark datasets



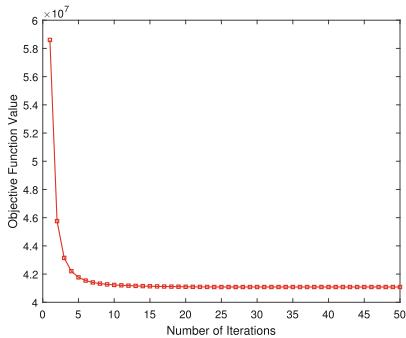
(a) YALE



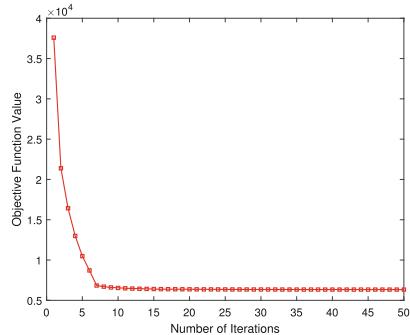
(b) JAFFE



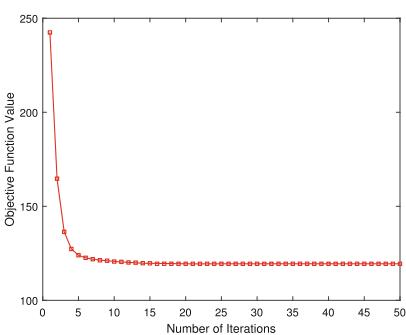
(c) ORL



(d) USPS



(e) LUNG



(f) COIL20

Fig. 3. The convergence curves of Algorithm 2 on six benchmark datasets.

be achieved when the value of p is small which could better approximates the $l_{2,0}$ sparsity constraint [15]. However, since the sparsity of the projection matrix is also affected by γ , and small p will make the problem highly nonconvex, the performance of FKMSp still requires joint parameter searching on practical datasets. Thus, we prepare to conduct more research on the deep impact of p on the model in the future. In summary, we suggest the finely parameter tuning for practical datasets.

In addition, we empirically evaluate the convergence of the FKMSp algorithm on benchmark datasets. The convergence curves of Algorithm 2 are depicted in Fig. 3. As can be observed, our algorithm is convergent, though not monotonic. This is mainly because Algorithm 1 is based on ALM, which is not a monotonically rising algorithm and the initial optimization direction depends heavily on the Lagrange multiplier and its penalty parameter.

6 Conclusion

In this paper, we presented an unsupervised feature selection model based on fuzzy K-Means and sparse projection (FKMSp). In particular, the proposed model can effectively reveal the data fuzziness by employing fuzzy K-Means to obtain soft-labels for data samples while measuring preserving the discriminative underlying structure by projection learning for feature selection. Additionally, an $l_{2,p}$ ($0 < p < 1$) norm constraint is imposed on projection matrix to effectively represent feature importance and enhance the model's robustness to the noises and outliers. We develop an efficient alternative optimization algorithm to solve the FKMSp model. The convergence of the algorithm were empirically validated though the objective function is not monotonically decreasing. The effectiveness and superiority of the FKMSp model on high-dimensional data clustering tasks are demonstrated by the theoretical analysis and experimental findings.

Acknowledgement. This work is supported by the Scientific Research Program Funded by Education Department of Shaanxi Provincial Government (No. 23JP107) and Shaanxi Province Key Research and Development Program (No. 2022ZDLSF07-07).

References

1. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, pp. 333–342. ACM, New York, NY, USA (2010). <https://doi.org/10.1145/1835804.1835848>
2. Celebi, M.E., Kingravi, H.A., Vela, P.A.: A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst. Appl. **40**(1), 200–210 (2013). <https://doi.org/10.1016/j.eswa.2012.07.021>
3. Chang, X., Wang, Q., Liu, Y., Wang, Y.: Sparse regularization in fuzzy c -means for high-dimensional data clustering. IEEE Trans. Cybern. **47**(9), 2616–2627 (2017). <https://doi.org/10.1109/TCYB.2016.2627686>

4. Chen, Q., Nie, F., Yu, W., Li, X.: $\ell_{2,p}$ -norm and Mahalanobis distance-based robust fuzzy c-means. *IEEE Trans. Fuzzy Syst.* **31**(9), 2904–2916 (2023). <https://doi.org/10.1109/TFUZZ.2023.3235384>
5. Havens, T.C., Bezdek, J.C., Leckie, C., Hall, L.O., Palaniswami, M.: Fuzzy c-means algorithms for very large data. *IEEE Trans. Fuzzy Syst.* **20**(6), 1130–1146 (2012). <https://doi.org/10.1109/TFUZZ.2012.2201485>
6. Huang, J., Nie, F., Huang, H.: A new simplex sparse learning model to measure data similarity for clustering. In: Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI 2015, pp. 3569–3575. AAAI Press (2015)
7. Kang, Z., Peng, C., Cheng, Q.: Robust PCA via nonconvex rank approximation. In: 2015 IEEE International Conference on Data Mining, pp. 211–220 (2015). <https://doi.org/10.1109/ICDM.2015.15>
8. Li, Z., Yang, Y., Liu, J., Zhou, X., Lu, H.: Unsupervised feature selection using nonnegative spectral analysis. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2012, pp. 1026–1032. AAAI Press (2012)
9. Liang, Y., Chen, Y., Huang, Q., Chen, H., Nie, F.: An effective optimization method for fuzzy k -means with entropy regularization. *IEEE Trans. Knowl. Data Eng.* 1–16 (2023). <https://doi.org/10.1109/TKDE.2023.3329821>
10. Liu, Y., Ye, D., Li, W., Wang, H., Gao, Y.: Robust neighborhood embedding for unsupervised feature selection. *Knowl.-Based Syst.* **193**, 105462 (2020). <https://doi.org/10.1016/j.knosys.2019.105462>
11. Lu, C., Lin, Z., Yan, S.: Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. *IEEE Trans. Image Process.* **24**(2), 646–654 (2015). <https://doi.org/10.1109/TIP.2014.2380155>
12. Nie, F., Hu, Z., Wang, X., Li, X., Huang, H.: Iteratively re-weighted method for sparsity-inducing norms. *IEEE Trans. Knowl. Data Eng.* **35**(7), 7045–7055 (2023). <https://doi.org/10.1109/TKDE.2022.3179554>
13. Nie, F., Huang, H., Cai, X., Ding, C.: Efficient and robust feature selection via joint $\ell_{2,1}$ -norms minimization. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, NIPS 2010, vol. 2, pp. 1813–1821 (2010)
14. Nie, F., Xue, J., Yu, W., Li, X.: Fast clustering with anchor guidance. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(4), 1898–1912 (2024). <https://doi.org/10.1109/TPAMI.2023.3318603>
15. Nie, F., Zhu, W., Li, X.: Structured graph optimization for unsupervised feature selection. *IEEE Trans. Knowl. Data Eng.* **33**(3), 1210–1222 (2021). <https://doi.org/10.1109/TKDE.2019.2937924>
16. Peizhuang, W.: Pattern recognition with fuzzy objective function algorithms (James C. Bezdek). *SIAM Rev.* **25**(3), 442 (1983). <https://doi.org/10.1137/1025116>
17. Ruspini, E.H.: Numerical methods for fuzzy clustering. *Inf. Sci.* **2**(3), 319–350 (1970). [https://doi.org/10.1016/S0020-0255\(70\)80056-1](https://doi.org/10.1016/S0020-0255(70)80056-1)
18. Wang, F., Zhu, L., Li, J., Chen, H., Zhang, H.: Unsupervised soft-label feature selection. *Knowl.-Based Syst.* **219**, 106847 (2021). <https://doi.org/10.1016/j.knosys.2021.106847>
19. Wang, J., Xie, F., Nie, F., Li, X.: Robust supervised and semisupervised least squares regression using $\ell_{2,p}$ -norm minimization. *IEEE Trans. Neural Netw. Learn. Syst.* **34**(11), 8389–8403 (2023). <https://doi.org/10.1109/TNNLS.2022.3150102>
20. Wang, R., Bian, J., Nie, F., Li, X.: Unsupervised discriminative projection for feature selection. *IEEE Trans. Knowl. Data Eng.* **34**(2), 942–953 (2022). <https://doi.org/10.1109/TKDE.2020.2983396>

21. Wang, S., Tang, J., Liu, H.: Embedded unsupervised feature selection. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 470–476. AAAI Press (2015)
22. Xu, J., Han, J., Xiong, K., Nie, F.: Robust and sparse fuzzy k-means clustering. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, pp. 2224–2230. AAAI Press (2016)
23. Xue, J., Nie, F., Wang, R., Li, X.: Iteratively reweighted algorithm for fuzzy c -means. *IEEE Trans. Fuzzy Syst.* **30**(10), 4310–4321 (2022). <https://doi.org/10.1109/TFUZZ.2022.3148823>
24. Yang, Y., Shen, H.T., Ma, Z., Huang, Z., Zhou, X.: $\ell_{2,1}$ -norm regularized discriminative feature selection for unsupervised learning. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, IJCAI 2011, vol. 2, pp. 1589–1594. AAAI Press (2011)
25. Zhang, R., Nie, F., Guo, M., Wei, X., Li, X.: Joint learning of fuzzy k-means and nonnegative spectral clustering with side information. *IEEE Trans. Image Process.* **28**(5), 2152–2162 (2019). <https://doi.org/10.1109/TIP.2018.2882925>
26. Zhao, W., Li, Q., Xu, H., Gao, Q., Wang, Q., Gao, X.: Anchor graph-based feature selection for one-step multi-view clustering. *IEEE Trans. Multimedia* **26**, 7413–7425 (2024). <https://doi.org/10.1109/TMM.2024.3367605>
27. Zuo, W., Meng, D., Zhang, L., Feng, X., Zhang, D.: A generalized iterated shrinkage algorithm for non-convex sparse coding. In: 2013 IEEE International Conference on Computer Vision, pp. 217–224 (2013). <https://doi.org/10.1109/ICCV.2013.34>



Open World Semi-supervised Learning Based on Multi-scale Enhanced Feature

Tianming Zhang¹, Kejia Zhang¹(✉), Haiwei Pan¹, and Yuechun Feng²

¹ School of Computer Science and Technology, Harbin Engineering University, Harbin, China

{ztm,kejiazhang,panhaiwei}@hrbeu.edu.cn

² School of Computer Science and Technology, Ningxia University, Yinchuan, China

Abstract. Semi-supervised learning (SSL) is one of the main approaches to address the high cost of manual annotation in supervised learning. In recent years, SSL methods have effectively utilized consistency regularization on unlabeled data to improve performance while leveraging a small portion of labeled data as anchors. A common assumption in most SSL methods is that the unlabeled data generally share similar distributions and structures. However, this assumption does not align with many real-world scenarios. We lack knowledge about the underlying data distribution of unlabeled data. Most existing SSL methods are limited in their applicability to real-world problems. In contrast, in this work, we aim to address the challenging problem of open-world SSL, where the objective is to identify samples belonging to known classes while detecting and clustering samples from novel classes that appear in the unlabeled data. This paper introduces a method to discover new classes based on comparing multi-scale enhancement features. Extensive experiments demonstrate that our method outperforms the current state-of-the-art methods on multiple popular classification benchmarks while providing a better trade-off between accuracy and training time.

Keywords: Open-world · Semi-supervised learning · Novel classes discover

1 Introduction

Deep neural networks have been proven effective in supervised learning tasks with a large amount of labeled data [9, 12, 14]. However, in the vast majority of real-world scenarios, data are not fully labeled, and annotating data requires substantial manpower. Semi-supervised learning (SSL) is one of the primary approaches to tackle this significant challenge, utilizing a large amount of unlabeled data along with a small portion of labeled data.

While recent SSL methods [7, 10, 13, 15, 18, 21, 25] have shown promising achievements, most of SSL methods assume that labeled and unlabeled data share the same underlying data distribution. Obviously, this assumption does

not hold in many real-world cases. The open-world problem involves not only classifying known samples but also clustering unknown samples.

In fact, the presence of samples from novel classes can significantly degrade the performance of standard SSL methods [16, 23]. This has led to the development of new methods [23, 24, 26] attempt to mitigate the impact of novel class samples by ignoring them. However, these approaches do not address the problem of classifying novel classes. Therefore, we seek a method that can both perform well in classifying known classes and accurately identify novel classes. This type of method is referred to as open-world semi-supervised learning, which was first introduced in ORCA [27]. We notice that existing open-world SSL methods [27, 28] suffer from significantly lower accuracy for novel classes compared to known classes, and although [29] manages to achieve comparable accuracy for novel classes, it comes at the cost of reduced accuracy for known classes.

Building upon [28, 29], we propose a method that discovers novel classes by utilizing the consistency between feature and semantic representations of samples after different image augmentations. This method does not require transforming the problem from open-world to closed-world [29]. Instead, it incorporates a generic novel class discovery module into the existing open-world SSL framework. This module leverages the high-dimensional feature and semantic similarity between labeled and unlabeled data to discover novel classes, with the high-dimensional feature and semantic labels of unlabeled data derived from the augmented versions of images used in the semi-supervised module. In addition, we define the concepts of soft pseudo-labels and hard pseudo-labels. The pseudo-labels generated in the semi-supervised module are categorized into soft pseudo-labels, representing the classification probabilities, and hard pseudo-labels, representing the highest probability value within the classification probabilities.

Two contributions are presented:

- In our work, we define the concepts of soft pseudo-labels and hard pseudo-labels. In the semi-supervised module, we categorize the generated pseudo-labels into soft and hard pseudo-labels. We find that by combining the soft or hard pseudo-labels with the labels of the labeled samples in the novel class discovery module, we can explore a wider range of features for novel classes.
- We propose to incorporate a novel class discovery module into the semi-supervised module to assist in addressing the problem of discovering and clustering new classes in open-world scenarios. This will be achieved by integrating enhanced features at multiple scales with the features in the novel class discovery module to discover novel classes.

The organization of our paper is presented as follows:

- Section 2 introduces open-world semi-supervised learning and related prior work.
- Section 3 defines the problem of semi-supervised learning and presents the two modules of our method: the semi-supervised module and the novel class discovery module.

- In Sect. 4, we conduct extensive experiments on three benchmark datasets and three additional fine-grained datasets, demonstrating that our proposed method outperforms existing works. Finally, we present experiments combining features and semantics from different image augmentations (Sect. 4.3), showing the benefits of the consistency-based approach in novel class discovery.

2 Related Work

2.1 Semi-supervised Learning

Semi-supervised learning (SSL) is a mature method for dealing with the bottleneck of label annotation in supervised learning [7, 10, 13, 15, 18, 21, 25]. Typically, SSL methods focus on the closed-world setting, where the unlabeled set contains only samples from known classes.

2.2 Closed-World SSL

Closed-world SSL methods primarily involve pseudo-labeling [7, 17, 20, 30] and consistency loss [10, 11, 13, 15]. Pseudo-labeling methods generate pseudo-labels for unlabeled samples, merge these samples with labeled samples, and then retrain the model. This process can be repeated multiple times, with an increasing number of unlabeled samples acquiring pseudo-labels until either no more unlabeled samples remain or the model converges. Consistency regularization methods aim to minimize the consistency loss between different versions of augmented images, thereby extracting meaningful features from unlabeled samples. Finally, hybrid methods [18, 21, 25] combine both consistency regularization and pseudo-labeling, with [18] and [25] being subsequently utilized in many SSL methods.

2.3 Open-World SSL

Open-world learning tackles the unconstrained nature of real-world data. In real-world scenarios, we cannot know the classification of previously unseen objects and need a method to assign them to novel classes. Recent research [16, 23] indicates that the presence of novel class samples degrades SSL performance. Robust SSL methods [23, 24, 26] address this issue by filtering or reweighting novel class samples.

TRSSL [28] generates pseudo-labels based on prior knowledge of the underlying class distribution, with the prior algorithm employing the Sinkhorn-Knopp algorithm. It also uses sharpening distributions for novel class clustering and applies a new uncertainty-aware temperature scaling technique to ensure reliable pseudo-labels. TRSSL performs well in various aspects, but the accuracy for novel classes remains slightly lower than that for known classes, prompting exploration of other novel class discovery and clustering methods. OpenLDN

[29] discovers novel classes in unlabeled data using labeled data as anchors and utilizes similarity loss for novel class discovery. OpenLDN exhibits a strong capability in discovering novel classes but may affect the accuracy of classification for labeled data, and it requires transforming the open-world problem into a closed-world problem and then solving it using closed-world SSL methods.

3 Methodology

In closed-world SSL methods, the assumption is that the unlabeled and labeled samples share the same underlying data distribution. However, this assumption does not hold for open-world SSL, where the focus is on how to discover novel classes and achieve satisfactory performance.

In the following subsections, we will discuss our approach to open-world SSL. We start by introducing the dataset configuration for open-world SSL in Sect. 3.1. Then, in Sect. 3.2, we describe the semi-supervised learning module. Finally, in Sect. 3.3, we present the novel class discovery module, which is a key component of our method to address the challenges of open-world SSL and achieve better performance for novel classes.

Table 1 shows some symbols used in the article, in which the upper corner marks 1 and 2 represent the image generated after different enhancement methods of the same image.

3.1 Problem Formulation

In the open-world SSL problem, the training set consists of a labeled set \mathbb{D}_L and an unlabeled set \mathbb{D}_U .

$\mathbb{D}_L = \{x_i^l, y_i^l\}_{i=1}^{N_l}$ represents the labeled dataset with N_l samples, where x_i^l is a labeled input sample, and y_i^l is its corresponding label, belonging to one of the C_L seen classes. $\mathbb{D}_U = \{x_i^u\}_{i=1}^{N_u}$ represents the unlabeled dataset with N_u ($N_u >> N_l$) samples, where x_i^u is an unlabeled input sample, belonging to one of the C_U unseen classes. Here, C_L is the total number of classes in \mathbb{D}_L and C_U is the total number of classes in \mathbb{D}_U . In the closed-world SSL setting, it is assumed that $C_L = C_U$, whereas in open-world SSL $C_L \subset C_U$. The reason for this distinction is that in open-world SSL, there are some samples from \mathbb{D}_U that do not belong to any of the seen classes. These samples belong to previously unseen classes, which we refer to as novel classes. Therefore, we assume C_N as the total number of novel classes, where $C_U = C_L + C_N$.

3.2 Semi-supervised Module and Pseudo-label Generation

The SSL part is illustrated in Fig. 1. Semi-Supervised Module. Similar to typical consistency regularization methods, we randomly apply a weak augmentation function $T_{aug}^1()$ to generate weakly augmented samples. We perform this augmentation on the unlabeled data $x_u \in \mathbb{D}_U$ to obtain two different augmented

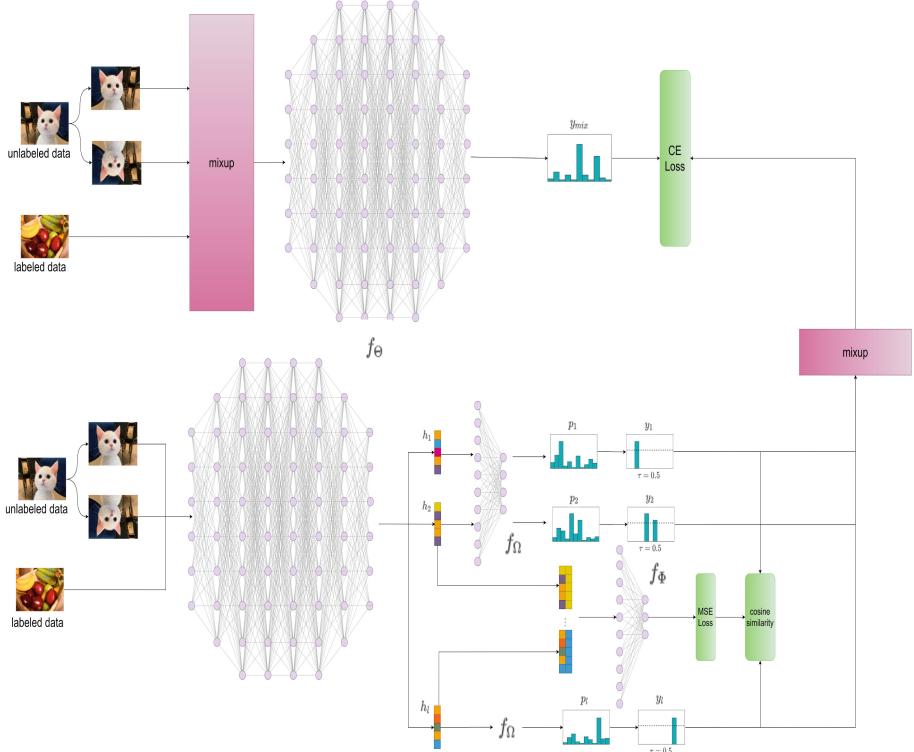


Fig. 1. Overview. The upper part is the Semi-Supervised Module, and the lower part is the Novel Class Discovery Module.

images: $\tilde{x}_u^1 = T_{aug}^1(x_u)$, $\tilde{x}_u^2 = T_{aug}^2(x_u)$. Here, \tilde{X}_u^1 represents the set of all \tilde{x}_u^1 in a batch, and \tilde{X}_u^2 represents the set of all \tilde{x}_u^2 in a batch.

Next, a convolutional neural network based encoder $f_\Theta()$ is used to extract high-dimensional feature information from these augmented images. Specifically, we compute $h_u^1 = f_\Theta(\tilde{x}_u^1)$, $h_u^2 = f_\Theta(\tilde{x}_u^2)$. Here, H_u^1 represents the set of all h_u^1 in a batch, and H_u^2 represents the set of all h_u^2 in a batch.

Then, We define soft and hard pseudo-labels. Soft pseudo-labels indicate the probability of classification we generate for the image, and hard pseudo-labels represent which category we think the image belongs to.

To generate soft pseudo-labels, a fully connected class prediction head f_Φ is employed to map the extracted features h to semantics. This mapping results in soft pseudo-labels $p_u^1 = f_\Phi(h_u^1)$, $p_u^2 = f_\Phi(h_u^2)$. We employ the Sinkhorn method [5] to obtain prior knowledge for soft pseudo-labels, $p_u^1 = \text{Sinkhorn}(p_u^1)$, $p_u^2 = \text{Sinkhorn}(p_u^2)$.

To obtain high-confidence hard pseudo-labels $\tilde{y}_u^1, \tilde{y}_u^2$ for soft pseudo-labels p_u^1, p_u^2 , We establish a threshold, denoted as τ . For unlabeled data, we selec-

Table 1. Some symbols used in the article.

Symbol	Instruction
\mathbb{D}_L	labeled dataset
\mathbb{D}_U	unlabeled dataset
N_l	total number of samples in the labeled dataset
N_u	total number of samples in the unlabeled dataset
C_L	number of classes in the labeled dataset
C_U	number of classes in the unlabeled dataset (total number of classes)
x_i^l, y_i^l	a sample from the labeled dataset
x_i^u	a sample from the unlabeled dataset
$\tilde{x}_u^1, \tilde{x}_u^2$	augmented images from the unlabeled data
$\tilde{X}_u^1, \tilde{X}_u^2$	set of augmented images from the unlabeled data in a batch
h_u^1, h_u^2	high-dimensional feature information of the unlabeled data
H_u^1, H_u^2	set of high-dimensional feature information of the unlabeled data in a batch
p_u^1, p_u^2	logits (soft pseudo-labels) of the unlabeled data
$\tilde{y}_u^1, \tilde{y}_u^2$	pseudo-labels (hard pseudo-labels) of the unlabeled data
$\tilde{Y}_u^1, \tilde{Y}_u^2$	set of pseudo-labels of the unlabeled data in a batch
X_m, Y_m	dataset after mixup
x_m, y_m	a sample from dataset after mixup

tively retain soft pseudo-labels that exceed this threshold, thereby obtaining hard pseudo-labels. Here, $\tilde{y}_u^1 = p_u^1(\max(p_u^1) \geq \tau)$, $\tilde{y}_u^2 = p_u^2(\max(p_u^2) \geq \tau)$.

Y_u^1 denotes the set of all \tilde{y}_u^1 in a batch, while Y_u^2 represents the set of all \tilde{y}_u^2 in a batch. We will elaborate on the application of the high-dimensional features, soft pseudo-labels, and hard pseudo-labels obtained here in Sect. 3.4.

We utilize mixup [18] as another form of image augmentation, with the specific blending scheme given by:

$$(X_m, Y_m) = \text{mixup}((\tilde{X}_u^1, Y_u^1), (\tilde{X}_u^2, Y_u^2), (X_L, Y_L)) \quad (1)$$

where, $(X_L, Y_L) \subseteq \mathbb{D}_L$ represents the set of labeled samples in the current batch.

In the image classification part of the semi-supervised module, the predicted labels for $\forall x_m \in X_m, \hat{y}_m = \text{softmax}(f_\Phi(f_\Theta(x_m)))$.

Finally, the loss for the semi-supervised module is obtained by combining the labels from both parts:

$$\mathcal{L}_{SS} = -\frac{1}{|X_m|} \sum_{x_m \in X_m} CE(y_m, \hat{y}_m) \quad (2)$$

where, CE denotes the cross-entropy, and $(x_m, y_m) \in (X_m, Y_m)$.

3.3 Novel Class Discovery

In the novel discovery module, we draw inspiration from the approach used in OpenLDN [29] to discover novel classes. This part is illustrated in Fig. 1. Novel Class Discovery Module. We combine feature similarity and semantic similarity to identify novel classes. The module calculates pairwise similarities between samples and clusters novel classes using labeled data as anchor points. We learn pairwise similarity using two-dimensional features. Therefore, the novel class discovery module can be divided into two parts: semantic similarity and feature similarity.

In this module, we employ a different approach from previous consistency regularization-based methods. When an image generates two different augmented images, all three images should belong to the same class, and therefore, the features generated from the two augmented images should be similar.

In Sect. 3.2, we obtained high-dimensional features h_u^1, h_u^2 , soft pseudo-labels p_u^1, p_u^2 , and hard pseudo-labels $\tilde{y}_u^1, \tilde{y}_u^2$ in the semi-supervised module. It is evident that these features or pseudo-labels should all come from the same class, and their final semantic classification should be the same. In the novel class discovery module, its input consists of high-dimensional features and logits from unlabeled data, and the results obtained from the semi-supervised module are used here. For instance, we utilize h_u^2 and \tilde{y}_u^1 as input elements for the novel class discovery module.

In [29], the approach utilized the features and pseudo-labels of the same image as inputs for novel class discovery. However, it is obvious that employing a consistency regularization method can lead to the learning of more diverse features compared to directly using a single image. Therefore, it becomes crucial to consider different ways of combining features or labels. For instance, using h_u^1, \tilde{y}_u^1 as inputs for the novel class discovery module, where they originate from the same augmented image, \tilde{x}_u^1 proves to be effective. However, this approach does not take into account a more extensive combination.

To address this limitation, we propose the concept of consistency regularization, which involves combining features and labels from different augmented images. For example, we consider using h_u^1, \tilde{y}_u^2 as inputs for the novel class discovery module. This combination allows for the learning of additional features. Certainly, the utilization of h_u^1, p_u^2 as inputs for the novel class discovery module is a viable option. The detailed combination approach will be presented in Sect. 4.3, where we provide a comprehensive analysis of the rationale behind the results obtained. For the purpose of convenient description, we adopt h_u^2, \tilde{y}_u^1 as examples to represent the inputs of the novel class discovery module in the following description.

Semantic Similarity. The semantic labels of unlabeled data and labeled data are used as inputs at the semantic layer of the novel class discovery module. We define $Y = Y_u^2 \cup Y_L$, where Y_L represents the set of labels from all labeled data in a batch.

Feature Similarity. The high-dimensional features of unlabeled data and labeled data are used as inputs at the feature layer of novel class discovery module. We define $H = H_u^2 \cup H_L$, where H_L represents the set of high-dimensional features from all labeled data in a batch.

The pairwise similarity loss for novel class discovery is defined as follows:

$$\mathcal{L}_{pair} = \sum_{(h', y'), (h', y'') \in (H, Y)} (Sim(y', y'') - f_\Omega(h', h''))^2 \quad (3)$$

where, $Sim()$ represents the cosine similarity function.

The final loss function \mathcal{L} is obtained by combining the semi-supervised module loss \mathcal{L}_{SS} and the pairwise similarity loss \mathcal{L}_{pair} :

$$\mathcal{L} = \mathcal{L}_{SS} + w\mathcal{L}_{pair} \quad (4)$$

where, w is the weight assigned to the pairwise similarity loss.

4 Experiments and Results

4.1 Experimental Setup

Dataset: We conducted experiments on three commonly used computer vision benchmark datasets: CIFAR-10 [2], CIFAR-100 [3], and Tiny ImageNet [6]. These datasets were selected in increasing order of difficulty based on the number of classes.

In the case of CIFAR-10, when using the first 50% of classes as seen classes and the remaining 50% of classes as novel classes, and using 10% of the data from the seen classes as the labeled set while the rest of the data are used as the unlabeled set, the labeled data for training would be 5% of the total dataset, which is 2,500 training samples. The remaining training samples are considered as unlabeled data.

In addition, we also conducted experiments on three other visual benchmark datasets: Oxford-IIIT Pet [4], FGVAircraft [8], and Stanford-Cars [6]. For all datasets, we used the first 50% of classes as seen classes and the remaining 50% of classes as unseen classes. To provide a detailed comparison with previous work, we primarily used 10% of the data from seen classes as the labeled set and the remaining 90% of data, including samples from unseen classes, as the unlabeled set for our experiments on standard benchmark datasets. However, there were also some experiments where we used 50% of the data from seen classes as the labeled set. Detailed results can be found in Sect. 4.2. The division of each data set is shown in Table 2.

Implementation Details. To ensure a fair comparison with previous works, we employ ResNet-18 [9] as the feature extractor f_Θ in our experimental process. We utilize a single hidden layer MLP with a dimension of 512 as the pairwise similarity prediction network f_Ω . The classifier f_Φ consists of multiple linear

Table 2. The division of each dataset

Dataset	No Class	Train Samples	Test Samples
CIFAR-10 [34]	10	50000	10000
CIFAR-100 [35]	100	50000	10000
Tiny ImageNet [38]	200	100000	10000
Oxford-IIIT Pet [53]	37	3680	3669
FGVAircraft [49]	100	6667	3333
Stanford-Cars [37]	196	8144	8041

layers, with l_2 normalization applied to the weights of the last linear layer. We consistently set the high-dimensional features in the novel class discovery module to be 512-dimensional. For all the datasets in our experiments, we train our model for 200 epochs. A batch size of 256 is used for the CIFAR-10, CIFAR-100, and Tiny ImageNet experiments, while a batch size of 128 is used for the other datasets. To optimize the network parameters, we employ the SGD optimizer with momentum. We adopt a linear warm-up learning rate schedule, starting with an initial learning rate of 0.1 and a warm-up length of 10 epochs. The weight decay is set to $1e-4$. Following [22], we set the value of λ to 0.05 and perform three iterations of pseudo-label generation using the Sinkhorn-Knopp algorithm. These settings ensure consistency and enable a fair comparison with previous works.

Evaluation Details. For evaluation, we report the standard classification accuracy on the seen classes, and the clustering accuracy on the novel classes, excluding the classes [11, 19, 26, 28, 29]. Due to the randomness introduced during dataset partitioning, all our reported results are obtained from experiments conducted on our pre-defined data splits.

For the novel classes, we treat the class predictions as cluster IDs. To assess the clustering accuracy, we employ the Hungarian algorithm [1] to match the cluster IDs with the ground truth classes. Once the matches are obtained, we calculate the classification accuracy based on the corresponding cluster IDs. Additionally, if a sample from a novel class is assigned to one of the seen classes, we consider it as a misclassification and remove that sample before matching the cluster IDs with the true class labels. We also report the clustering accuracy for all classes.

By reporting these metrics, we provide a comprehensive evaluation of both the classification accuracy on seen classes and the clustering accuracy on novel classes, enabling a thorough assessment of the model’s performance.

4.2 Result

In comparison to the literature on the problem of open-world SSL, we present a thorough comparative analysis of our approach. The benchmark dataset, com-

Table 3. CIFAR-10, CIFAR-100, and Tiny Imagenet, 50% seen, 10% labeled.

Method	CIFAR-10			CIFAR-100			Tiny ImageNet		
	Known	Seen	All	Known	Seen	All	Known	Seen	All
DTC [23]	42.7	31.8	32.4	22.1	10.5	13.7	13.5	12.7	11.5
RankStats [22]	71.4	63.9	66.7	20.4	16.7	17.8	9.6	8.9	6.4
UNO [17]	86.5	71.2	78.9	53.7	33.6	42.7	28.4	14.4	20.4
OpenLDN-mixmatch [7]	92.6	93.6	93.2	55.0	40.0	47.7	32.6	7.3	20.2
TRSSL [16]	94.0	87.2	90.6	66.6	45.7	56.2	38.2	19.8	28.8
Ours	95.5	93.6	95.2	66.5	51.5	58.4	37.6	20.5	29.4

monly utilized in this context, involves utilizing a mere 5% of the total training samples as labeled data. The dataset, herein referred to as the baseline dataset, is detailed in Table 3. This dataset encompasses **CIFAR-10**, **CIFAR-100**, and **Tiny Imagenet**, each partitioned such that 50% of the classes are designated as seen, and the remaining 50% as novel. Within the set of seen classes, 10% of the data is utilized as the labeled subset, while the remainder forms the unlabeled subset. The outcomes presented herein are the average results across three independent experimental runs.

In evaluating the performances of various methodologies, we meticulously reproduce the procedures of TRSSL [28] and OpenLDN [29]. Subsequently, a comprehensive juxtaposition of their results with those of our own method is undertaken. Our approach, when assessed on the CIFAR-10 dataset, notably surpasses the accuracy in novel class identification achieved by ORCA [27] by a margin of 7.9%, and surpasses the performance of TRSSL [28] by 4.0%. Particularly noteworthy is the remarkable efficacy demonstrated by our method in the context of novel class discovery when the total number of classes is relatively modest. Additionally, the discernible impact on the accuracy of the seen classes is comparatively minimal.

Evidently, our method exhibits robust performance under conditions of limited labeled data, particularly in scenarios with a restricted number of classes. However, as the overall number of classes escalates to 200, the simultaneous optimization of novel class discovery and the preservation of seen class accuracy becomes a challenging endeavor.

Table 4. Oxford-IIIT Pet, FGVC-Aircraft, and Stanford-Cars, 50% seen, 10% labeled.

Method	Oxford-IIIT Pet			FGVC-Aircraft			Stanford-Cars		
	Known	Seen	All	Known	Seen	All	Known	Seen	All
OpenLDN-mixmatch	40.1	18.8	29.4	25.3	16.7	21.7	8.2	8.4	8.3
TRSSL	41.7	33.6	37.6	25.0	30.4	27.7	11.4	11.7	11.5
Ours	40.9	37.0	39.8	26.1	31.4	28.7	10.4	12.3	10.9

In this context, specific experiments have been conducted for TRSSL [28] and OpenLDN [29], following the partitioning methodology we employ for the Oxford-IIIT Pet, FGVAircraft, and Stanford-Cars datasets. As such, we have undertaken an independent replication of the methodologies proposed by TRSSL [28] and OpenLDN [29] using our own experimental setup, thereby enabling a comprehensive comparative analysis against our proposed approach.

Table 4 presents the average accuracy achieved on the Oxford-IIIT Pet, FGVAircraft, and Stanford-Cars datasets, where the class distribution entails 50% of the classes as seen and the remaining 50% as novel. Akin to the aforementioned schema, 10% of the data from the seen classes are earmarked as the labeled subset, while the residual data constitute the unlabeled counterpart.

Table 5. CIFAR-10, CIFAR-100, and Tiny Imagenet, 50% seen, 50% labeled.

Method	CIFAR-10			CIFAR-100			Tiny ImageNet		
	Known	Seen	All	Known	Seen	All	Known	Seen	All
OpenLDN-mixmatch	95.2	92.7	93.9	75.3	46.8	61.2	52.3	19.5	25.9
OpenLDN-UDA	95.7	95.1	95.4	74.1	44.5	59.3	58.3	25.5	41.9
TRSSL	96.8	92.8	94.8	80.2	49.3	64.0	59.1	24.2	42.1
Ours	96.8	88.6	92.7	80.9	47.5	63.3	57.4	26.5	41.9

Table 6. Oxford-IIIT Pet, FGVAircraft, and Stanford-Cars 50% seen, 50% labeled.

Method	Oxford-IIIT Pet			FGVC-Aircraft			stans_cars		
	Known	Seen	All	Known	Seen	All	Known	Seen	All
DTC [23]	20.7	16.0	13.6	16.3	16.5	11.8	12.3	10.0	7.7
RankStats [22]	12.6	11.9	11.1	13.4	13.6	11.1	10.4	9.1	6.6
UNO [17]	49.8	22.7	34.9	44.4	24.7	31.8	49.0	15.7	30.7
OpenLDN	63.1	31.0	47.5	60.1	30.4	45.7	50.9	25.6	38.7
TRSSL	77.3	46.1	63.8	62.5	32.5	47.5	69.0	33.0	51.0
Ours	77.6	48.8	65.1	64.7	42.4	53.5	65.1	31.7	48.4

It is evident that our method achieves significant improvements when using only 10% of the seen-class data as the labeled set. Our approach outperforms other open-world semi-supervised learning methods in terms of accuracy for unseen classes. Furthermore, when utilizing 50% of the seen-class data as the labeled set, our method demonstrates excellent performance on datasets with a relatively small number of categories such as Oxford-IIIT Pet and FGVC-Aircraft. Table 5 displays the performance when using 50% of the seen-class data

as the labeled set on CIFAR-10, CIFAR-100, and Tiny ImageNet and Table 6 displays the performance when using 50% of the seen-class data as the labeled set on Oxford-IIIT Pet, FGVAircraft, and Stanford-Cars. It can be observed that the ability to discover new classes diminishes as the amount of labeled data increases, particularly in scenarios with a smaller number of categories.

4.3 Ablation and Analysis

In the module dedicated to novel class discovery, we placed particular emphasis on exploring the use of consistency regularization to obtain more diverse features. Experimental results have demonstrated the significant effectiveness of this approach. To this end, we compared the combination of different-dimensional features from the same image, such as h_u^1, p_u^1 , as the baseline, with the combination of different-dimensional features from the same image after different augmentations such as h_u^1, p_u^2 . Additionally, during the experiments, we discovered that in scenarios involving different datasets or varying proportions of labeled data, sometimes employing more precise hard pseudo-labels yielded superior results. Subsequently, we also present the combinations h_u^1, \tilde{y}_u^1 , as well as h_u^1, \tilde{y}_u^2 .

From the results in Table 7, it can be observed that the performance on the Cifar-10 dataset is better when using more accurate hard pseudo-labels compared to soft pseudo-labels. Conversely, on the Cifar-100 dataset, the performance is superior when using soft pseudo-labels instead of hard pseudo-labels. This outcome is evidently influenced by the quality of the dataset itself. Cifar-100 has only one-tenth the number of images per category compared to Cifar-10 during training, hence utilizing soft pseudo-labels, which are less affected by confirmation bias, enables the learning of more diverse features.

Table 7. The effects of different inputs in the novel class discovery module

Used	CIFAR-10			CIFAR-100		
	Known	Seen	All	Known	Seen	All
h_u^1, p_u^1	90.4	93.1	91.7	65.8	48.8	57.3
h_u^1, \tilde{y}_u^1	94.5	93.6	94.2	62.4	42.6	55.9
h_u^1, p_u^2	91.0	93.8	92.4	66.5	51.5	58.4
h_u^1, \tilde{y}_u^2	95.7	94.6	95.2	65.9	48.4	57.2

There is a parameter setting that requires special consideration, which is the threshold τ used when transitioning from soft pseudo-labels to hard pseudo-labels. The value of τ determines the confidence level we attribute to these pseudo-labels. A higher τ value can help mitigate the impact of confirmation bias. However, as τ becomes larger, it may diminish the effect on the accuracy of familiar classes, potentially weakening the ability to discover novel classes. The detailed results are presented in Table 8. Here CIFAR-10 uses the hard

pseudo-label as the input of novel class discovery, and CIFAR-100 uses the soft pseudo-label as the input of novel class discovery.

Table 8. The influence of different threshold values τ on the accuracy.

τ	Cifar-10			Cifar-100		
	Known	Seen	All	Known	Seen	All
0.3	93.3	91.3	92.3	66.0	42.6	54.3
0.5	95.7	94.6	95.2	66.5	51.5	59.0
0.7	94.4	94.2	94.3	67.2	46.6	57.9
0.95	95.9	90.4	93.1	68.5	47.1	57.0

5 Conclusion

In this work, we propose the a new method to address the open-world Semi-Supervised Learning (SSL) problem. Our method extends the existing open-world semi-supervised methods by incorporating a novel-class discovery module to assist in the discovery and clustering of novel classes in open-world scenarios. We observe that the integration of the novel-class module with the semi-supervised module enables the exploration of a wider range of novel class features by leveraging the diverse features of augmented images and the semantic consistency. Through empirical experiments, we provide evidence that the performance of different modules can be enhanced by adhering to the principle of consistency. To validate the effectiveness of our proposed method, we conduct extensive experiments on six distinct datasets. The results of these experiments conclusively demonstrate the efficacy of our approach in addressing the challenges associated with discovering and clustering novel classes in open-world scenarios. Notably, our method exhibits superiority over existing solutions in terms of its ability to accurately identify and cluster novel classes.

Acknowledgment. The work was supported by Ningxia Natural Science Foundation Project (2022AAC03346), the National Natural Science Foundation of China under Grant No. 62072135.

References

1. Kuhn, H.W.: The Hungarian method for the assignment problem. *Naval Res. Logistics Q.* **2**(1–2), 83–97 (1955)
2. Krizhevsky, A., Nair, V., Hinton, G.: CIFAR-10. *Can. Inst. Adv. Res.* **5**, 4 (2009)
3. Krizhevsky, A., Nair, V., Hinton, G.: CIFAR-10 (Canadian Institute for Advanced Research), vol. 5(4), p. 1 (2010). <http://www.cs.toronto.edu/kriz/-cifar.html>

4. Parkhi, O.M.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3498–3505. IEEE (2012)
5. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
6. Krause, J., et al.: 3D object representations for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 554–561 (2013)
7. Lee, D.-H., et al.: Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3(2), p. 896, Atlanta (2013)
8. Maji, S., et al.: Fine-grained visual classification of aircraft. arXiv preprint [arXiv:1306.5151](https://arxiv.org/abs/1306.5151) (2013)
9. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. en-US. In: arXiv Neural and Evolutionary Computing, October 2016
11. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
12. He, K., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
13. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. en-US. In: International Conference on Learning Representations, January 2017
14. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
15. Miyato, T., et al.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1979–1993 (2018)
16. Oliver, A., et al.: Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
17. Shi, W., Gong, Y., Ding, C., Ma, Z., Tao, X., Zheng, N.: Transductive semi-supervised deep learning using min-max features. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 311–327. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_19
18. Berthelot, D., et al.: MixMatch: a holistic approach to semi-supervised learning. en-US. arXiv Learning, May 2019
19. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8401–8409 (2019)
20. Arazo, E., et al.: Pseudo-labeling and confirmation bias in deep semisupervised learning. en-US. In: 2020 International Joint Conference on Neural Networks (IJCNN), July 2020. <https://doi.org/10.1109/ijcnn48605.2020.9207304>
21. Berthelot, D., et al.: ReMixMatch: semi-supervised learning with distribution matching and augmentation anchoring. en-US. In: International Conference on Learning Representations, International Conference on Learning Representations, April 2020

22. Caron, M., et al.: Unsupervised learning of visual features by contrasting cluster assignments. *Adv. Neural. Inf. Process. Syst.* **33**, 9912–9924 (2020)
23. Chen, Y., et al.: Semi-supervised learning under class distribution mismatch. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3569–3576 (2020)
24. Guo, L.-Z., et al.: Safe deep semi-supervised learning for unseen-class unlabeled data. In: International Conference on Machine Learning. PMLR, pp. 3897–3906 (2020)
25. Sohn, K., et al.: FixMatch: simplifying semi-supervised learning with consistency and confidence. en-US. In: Cornell University - arXiv, January 2020
26. Zhao, X., et al.: Robust semi-supervised learning with out of distribution data. arXiv preprint [arXiv:2010.03658](https://arxiv.org/abs/2010.03658) 2.3 (2020)
27. Cao, K., Brbic, M., Leskovec, J.: Open-world semi-supervised learning. arXiv preprint [arXiv:2102.03526](https://arxiv.org/abs/2102.03526) (2021)
28. Rizve, M.N., Kardan, N., Shah, M.: Towards realistic semi-supervised learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. ECCV 2022. LNCS, vol. 13691, pp. 437–455. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19821-2_25
29. Rizve, M.N., Kardan, N., Khan, S., Shahbaz Khan, F., Shah, M.: OpenLDN: learning to discover novel classes for open-world semi-supervised learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. ECCV 2022. LNCS, vol. 13691, pp. 382–401. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19821-2_22
30. Moezzi, M.: An uncertainty-aware pseudo-label selection framework using regularized conformal prediction. arXiv preprint [arXiv:2309.15963](https://arxiv.org/abs/2309.15963) (2023)



ACD: Attention Driven Cognitive Diagnosis for New Learners Joining ITS

Bingdi Shao^{1,4}, Keai Wei^{1,4}, Longjiang Guo^{1,2,3,4(✉)}, Meirui Ren^{1,2,3,4},
Lichen Zhang^{1,2,3,4(✉)}, and Peng Li^{1,2,3,4}

¹ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
{longjiangguo,zhanglichen}@snnu.edu.cn

² Key Laboratory of Intelligent Computing and Service Technology for Folk Song,
Ministry of Culture and Tourism, Xi'an 710119, China

³ Key Laboratory of Modern Teaching Technology, Ministry of Education,
Xi'an 710062, China

⁴ Engineering Laboratory of Teaching Information Technology of Shaanxi Province,
Xi'an 710119, China

Abstract. In Intelligent Tutor Systems (ITS), Cognitive Diagnosis (CD) is an important and fundamental problem, which aims to discover learners' proficiency in different knowledge concepts. However, existing CD models (CDMs) that are from the perspective of learners and scores ignore the cold start problem of new learners joining ITS (CSP for short). This paper proposes an attention mechanism-driven cognitive diagnosis model named ACD for new learners joining ITS to solve the cold start problem, which is composed of a three-layer attention mechanism neural network. Specifically, in the first layer, the cognitive state of part of the concepts was obtained using the attention mechanism on the learners' exercises, concepts, and scores. In the second layer, attention is computed on all concepts and on the part of the cognitive state on concepts output in the first layer to obtain the cognitive state on all concepts. In the third layer, concepts, exercises, and the cognitive state of the learner output in the second layer was obtained using the attention mechanism on the learners' scores on the exercises. Finally, the large number of experimental results on five real datasets show that ACD performs well on different evaluation metrics when new learners come into ACD.

Keywords: Cognitive diagnosis · Education · Attention mechanism · Cold start

1 Introduction

In Intelligent Tutor Systems (ITS) [1], how to make more accurate real-time learning diagnoses and intelligent scoring is the focus of researchers. As the core technology of personalized learning [20], the task of the cognitive diagnosis

B. Shao and K. Wei—Contributed equally to this work.

model (CDM) is to predict learners' mastery of related concepts based on their learning activity records over time. A toy example of CDM is shown in Fig. 1. In the practice process of learners, the scores in the practice are used as the input of the CDM, and the output value is the cognitive state of learners. Based on these cognitive states, we can predict the scores of learners [8], recommend learning paths [25] and learning earlier warning [4], and so on.

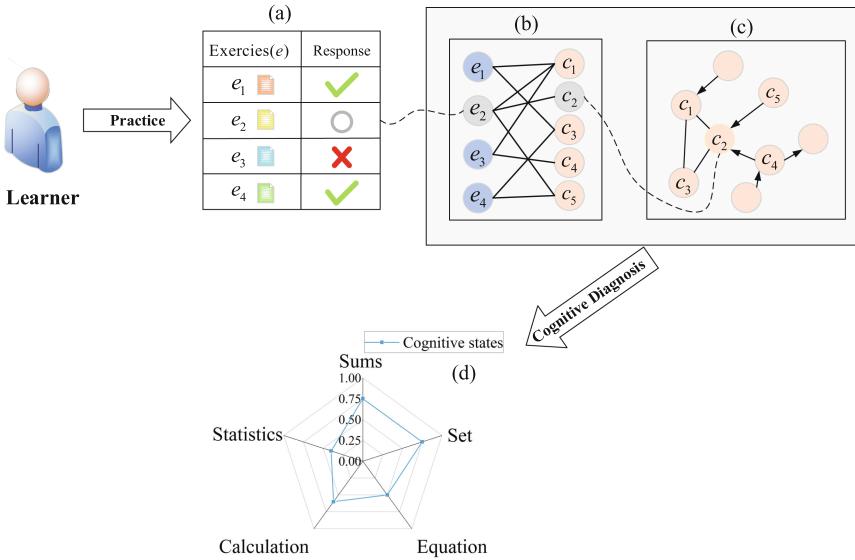


Fig. 1. A toy example of cognitive diagnosis. (a) Record of the learner's response, and “O” indicating not answered; (b) The interrelationship between the exercise and the concept of knowledge; (c) The interrelationship between knowledge concepts; (d) Cognitive state of learners.

Existing CDMs took learners' historical exercise records as inputs for deep training and predicted whether learners can finish new exercises correctly. When new learners join ITS, the trained CDMs did not master the learning characteristics of the new learners. Thus, the accuracy of existing CDMs decreases when predicting the exercise scores of new learners. This paper experimentally verifies this fact (see Table 3 and Table 4). This fact is called **the cold start problem of new learners (CSP for short)**.

Earlier CDMs mainly used probabilistic statistics and educational psychology methods, among which Item Response Theory (IRT) [6,14] and Deterministic Inputs, Noisy “and” Gate (DINA) [9] are two classical CDMs based on educational psychology theory. IRT is represented by three scalars respectively in terms of project difficulty, differentiation, and learners' ability. IRT modeled and analyzed the probability of learners' correct answers. However many mental structures are multidimensional, which makes it possible that the one-dimensional

limitations of IRT need to be overcome. Reckase et al. [16] extended IRT to the multi-dimensional item response theory (MIRT), using multi-dimensional hiding ability to depict learner states. However, both IRT and MIRT ignore the CSP.

DINA [9] uses 0 or 1 to indicate whether the learner has mastered the concept of knowledge, and DINA assumes the question can only be answered correctly if all the concepts of the exercise are mastered. At the same time, DINA proposes two parameters, **slip** and **guess**, which are equivalent to two noises. Learners need to master all the necessary skills involved in an item. If learners lack at least one of the required skills, they need to **guess** and choose the correct answer. DINO [18] redefined the “compensator” mode of action of knowledge capabilities in DINA. Jimmy de La et al. [10] investigated the effect of higher-order attribute parameters on the diagnostic effect and proposed the Generalized Deterministic Inputs, Noisy “and” Gate (G-DINA). Compared to these CDMs, DINA involves fewer parameters is easier to interpret, and has been studied quite a bit by researchers. Tong et al. [19] proposed Item Response Ranking (IRR), in which assumes that learners’ proficiency is monotonic and the probability of giving correct responses to items is also monotonic. To consider the problem of different distributions of **slip** and **guess** factors, Liu et al. [13] proposed FuzzyCDF, which firstly fuzzifier the skill proficiency of candidates based on the fuzzy set assumption. Cheng et al. [3] proposed a simple but effective Deep Item Response Theory (DIRT) framework, which DIRT can be used to enhance the semantic mining process, including three modules: input, deep diagnosis, and intervention. Again, the CSP has been not considered in above CDMs.

To address the problems of traditional cognitive diagnostic functions due to the limitations of manual design and the difficulty of traditional CDMs to mine item text content information, Wang et al. [22] was the first one to propose Neural Cognitive Diagnosis (NeuralCD). It combined cognitive diagnosis with deep learning methods, employed neural networks to learn complex nonlinear item interactions, learners, and items as factor vectors, and modeled their interactions using multiple neural layers, which could enhance diagnostic results and make them interpretable. To address the problem of modeling learner and knowledge skills as inter-layer interactions, Gao et al. [8] proposed the Relationship Diagram Driven Cognitive Diagnosis (RCD) framework, which explicitly integrates interactive and structured information with multiple relationships by building a hierarchical “Learner-Item-Concept” relationship diagram unified model, a scalable diagnostic function to predict learners performance and co-train networks. Wang et al. [23] proposed CDGK, which not only could capture the nonlinear interactions between practice features, learner scores, and knowledge concepts, but also reduce the dimensionality of the model without losing accuracy. The CSP has been not considered in above CDMs.

In addition, since the rapid development of deep learning is widely used in the fields of natural language processing, statistical learning, image detection, speech recognition, and computers, attention mechanism is the core technology [2]. Vaswani et al. [21] proposed the Transformer architecture that completely eliminates sequential processing and repeated connections, which relies only on

the self-attentive mechanism to capture global dependencies between inputs and outputs. The method has a few advantages, such as a higher significant parallel processing, a shorter training time, and a higher accuracy. In the field of cognitive diagnosis, many researchers introduced attentional mechanisms to solve the interaction problem of concepts. Gao et al. [7] proposed DeepCDM. It modeled learners' problem mastery based on attentional mechanisms and neural networks while considering the importance and interaction of skills. However, DeepCDM is only applicable to small-scale datasets, and the interaction between concepts depends on the keyword text of the concepts, so it is not applicable to ITS. Again, the CSP has been not considered in above CDMs.

To solve the CSP, this paper proposes ACD. It introduces an attention mechanism to capture the deep and complex relationships. That is the relationship of exercises and knowledge concepts and the relationship of knowledge concepts and knowledge concepts. Specifically, ACD is a neural network consisting of three layers of attention mechanisms. By considering the interdependencies between items and using the weights assigned to the relevant items as prediction targets, ACD can capture the complex interactions between exercises and concepts and concepts and concepts, thus providing a more accurate picture of the learners' cognitive state.

The innovative points of this paper are as follows:

- A attention mechanism driven cognitive Diagnosis model (ACD) for new coming learners is proposed. In ACD, the complex quantitative relationships between exercises and concepts, and concepts and concepts are calculated through the attention mechanism. The parameters of each layer outputs are between 0 and 1, which are interpretable.
- The CSP is solved. It has been not considered in previous CDMs. The results show that ACD outperforms both the existing classical CDMs (DINA, MIRT) and the latest cognitive diagnostic CDMs (CDGK, NCDM and RCD) in terms of performance and interpretability.

2 Problem Formulation

Before the definition of the problem, this section gives some notations needed for the different symbols. Non-bold uppercase letters are used to represent sets, such as L ; bold capital letters indicate matrices, such as \mathbf{Q} ; bold lowercase letters indicate vectors, such as \mathbf{r} ; the set is represented by a non-bold italic capital letter, such as E ; numbers are represented in cursive capital letters, such as \mathcal{G} . The important symbols used in this paper are shown in Table 1.

Hypothesis 1: According to the monotonicity hypothesis [17], the more proficient a learner is with the concepts contained in an exercise, the higher the learner's score on that exercise.

Table 1. Important symbols used in this paper

Symbol	Description
L	Learners, $ L = \mathcal{N}$, l_i is the i -th learner in the dataset
E	Training set, $ E = \mathcal{M}$, e_m is the m -th exercise in the dataset
C	Knowledge concept set, $ C = \mathcal{K}$, c_k is the k -th knowledge concept in the dataset
\mathbf{Q}	Matrix of relationships between exercises and concepts, $\mathbf{Q} \in \mathbb{Z}^{\mathcal{M} \times \mathcal{K}}$
\mathbf{P}	Quantitative relationship matrix between exercises and concepts, $\mathbf{P} \in \mathbb{R}^{\mathcal{M} \times \mathcal{K}}$
\mathbf{T}	Quantitative interaction matrix between concepts, $\mathbf{T} \in \mathbb{R}^{\mathcal{K} \times \mathcal{K}}$
\mathbf{P}''	Exercises embedding matrix, \mathbf{p}''_m denotes the m -th row of $\mathbf{P}'' \in \mathbb{R}^{\mathcal{M} \times \mathcal{H}}$, p''_{mh} is an element in the \mathbf{p}''_m , when initialized, $p''_{mh} \sim N(0, 1)$, \mathcal{H} is a hyperparameter
\mathbf{T}''	Concepts embedding matrix, \mathbf{t}''_k denotes the i -th row of $\mathbf{T}'' \in \mathbb{R}^{\mathcal{K} \times \mathcal{H}}$, t''_{kh} is an element in the \mathbf{t}''_k , when initialized, $t''_{kh} \sim N(0, 1)$
\mathbf{A}	The cognitive state of all learners, $\mathbf{A} \in \mathbb{R}^{\mathcal{N} \times \mathcal{K}}$
\mathbf{a}_i	Row i of \mathbf{A} , denotes the cognitive state $\mathbf{a}_i \in \mathbb{R}^{1 \times \mathcal{K}}$ of learner l_i
E_i	Learner l_i has answered the exercise
R_i	The answer record of l_i in E_i
$E_i^{(X)}$	The cognitive state of learner l_i in part E_i
X_i	The score of learner l_i for the exercises in $E_i^{(X)}$
$E_i^{(Y)}$	Another part of the E_i , $E_i^{(X)} \cup E_i^{(Y)} = E_i$
Y	The real score of l_i in $E_i^{(Y)}$, $X_i \cup Y_i = R_i$
\hat{Y}_i	The score of learner l_i for the exercises in $E_i^{(Y)}$
\mathbf{s}	Learners' slip rate in each exercise, $\mathbf{s} \in \mathbb{R}^{1 \times \mathcal{M}}$
\mathbf{g}	Guess rate of learners on each exercise, $\mathbf{g} \in \mathbb{R}^{1 \times \mathcal{M}}$
i, v	They are variables and are often used as subscripts
$ \cdot $	If the element in $ \cdot $ is a scalar, it means taking the absolute value, and if the element is a set, it means taking the number of elements in the set

Hypothesis 2: Learners' scores on exercises are mainly influenced by the concepts included in the exercise, and different concepts have different effects on the exercise [24].

Theory 1: According to pedagogical theory [5, 11, 15], there is an interaction between concepts, i.e., learners' proficiency in one concept is influenced by the other concepts.

Suppose there are \mathcal{N} learners, \mathcal{M} exercises, and \mathcal{K} knowledge concepts in ITS, which can be denoted as $L = \{l_1, l_2, \dots, l_{\mathcal{N}}\}$, $E = \{e_1, e_2, \dots, e_{\mathcal{M}}\}$, and $C = \{c_1, c_2, \dots, c_{\mathcal{K}}\}$, respectively. $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{\mathcal{N}}\}$ denotes the cognitive state of all learners, and $\mathbf{a}_i = \{a_{i1}, a_{i2}, \dots, a_{i\mathcal{K}}\}$ denotes the cognitive state of the i -th learner in \mathbf{A} , where $(0 \leq a_{ik} \leq 1)$. $\mathbf{A}^{\circ} = \{\mathbf{a}_1^{\circ}, \mathbf{a}_2^{\circ}, \dots, \mathbf{a}_{\mathcal{N}}^{\circ}\}$ denotes the cognitive state of all new learners, $\mathbf{a}_i^{\circ} = \{a_{i1}^{\circ}, a_{i2}^{\circ}, \dots, a_{i\mathcal{K}}^{\circ}\}$ denotes the cognitive state of the i -th new

learner, where $(0 \leq a_{ik}^{\circ} \leq 1)$. The inclusion relationship between the exercise and the concept is represented as $\mathbf{Q} = (q_{mk}) \in \mathbb{Z}^{\mathcal{M} \times \mathcal{K}}$, where element $q_{mk} \in \{0, 1\}$. The exercise record of learner l_i is denoted as $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{i\mathcal{M}}\} \in \mathbb{R}^{1 \times \mathcal{M}}$, where $x_{im} (0 \leq x_{im} \leq 1)$ denotes the regularized result of learner l_i 's score on exercise e_m .

Problem Statement: Suppose an ITS system with a cognitive diagnostic model parameterized by \mathbf{A} . The goal is to predict the cognitive state \mathbf{A}_i° for an arbitrary new learner $l_i^{\circ} \notin L$ with a small number of practice records E_i° .

3 ACD Model

This section describes a novel neural network framework designed to model ACD, which can adapt to different lengths of answer records and take advantage of quantitative interactions between concepts and quantitative relationships between exercises and concepts. The ACD model is also designed with slip and guess to make it more interpretable.

The core ACD idea is as follows: first, ACD uses the idea of cross-validation to divide the answer records R_i of l_i into X_i and Y_i , where R_i is from the training set. Then, ACD obtains the cognitive state \mathbf{a}_i^* of the learner's partial concept based on X_i , \mathbf{P}'' and \mathbf{T}'' . Then, the cognitive state \mathbf{a}_i of the learner's partial concept is obtained based on \mathbf{a}_i^* , \mathbf{T}'' . Next, the score \hat{Y}_i of l_i is predicted based on \mathbf{a}_i , **slip**, **guess**, \mathbf{P}'' and \mathbf{T}'' . Finally, \hat{Y}_i is computed with the actual score Y_i of l_i for *Loss*, and the parameters of ACD are updated by the back propagation algorithm **Adam** [12]. Thus, after a certain number of iterations, the model can predict the score in another part of the exercise based on a small number of practice records of the new learner.

3.1 Divide Answer Records to Enhance the Data

A learner only answers a few of the exercises, and all learners do not necessarily answer the same number of questions. As shown in Fig. 2, there are 10 exercises, so $\mathcal{M} = 10$, and the learner answered 9 of them, $\mathbf{E}_i = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$. Since the scores of one part of l_i 's exercises are used to predict the scores of the other part of the exercises. This paper divides l_i 's answer record R_i into two parts: X_i and Y_i . As shown in Fig. 2, the exercise record $R_i = \{r_{i1} = 1.00, r_{i2} = 0.32, r_{i3} = 0.83, r_{i4} = 0.63, r_{i5} = 0.47, r_{i6} = 0.59, r_{i7} = 0.70, r_{i8} = 0.95, r_{i9} = 0.00\}$ of l_i , is randomly divided into α ($\alpha = 3$) parts: $\{r_{i2}, r_{i4}, r_{i9}, \}$, $\{r_{i3}, r_{i6}, r_{i8}, \}$ and $\{r_{i1}, r_{i5}, r_{i7}, \}$. Each part is treated as Y_i and the remaining parts are combined as X_i . Thus, R_i is augmented with α response records:

$$\begin{aligned} & \{X_i = \{x_{i1} = r_{i1}, x_{i2} = r_{i3}, x_{i3} = r_{i5}, x_{i4} = r_{i6}, x_{i5} = r_{i7}, x_{i6} = r_{i8}\}, Y_i = \{y_{i1} = r_{i2}, y_{i2} = r_{i4}, y_{i3} = r_{i9}\}\}; \\ & \{X_i = \{x_{i1} = r_{i1}, x_{i2} = r_{i2}, x_{i3} = r_{i4}, x_{i4} = r_{i5}, x_{i5} = r_{i7}, x_{i6} = r_{i9}\}, Y_i = \{y_{i1} = r_{i3}, y_{i2} = r_{i6}, y_{i3} = r_{i8}\}\}; \end{aligned}$$

$$\{X_i = \{x_{i1} = r_{i2}, x_{i2} = r_{i3}, x_{i3} = r_{i4}, x_{i4} = r_{i6}, x_{i5} = r_{i8}, x_{i6} = r_{i9}\}, Y_i = \{y_{i1} = r_{i1}, y_{i2} = r_{i5}, y_{i3} = r_{i7}\}\}.$$

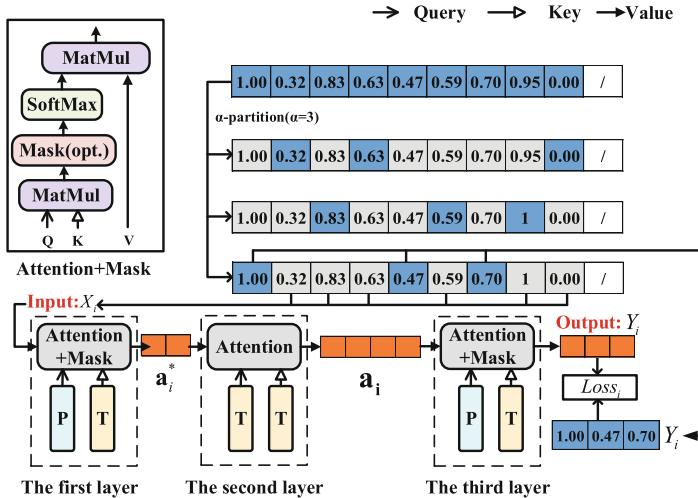


Fig. 2. The framework of ACD.

3.2 ACD Based on Attention Mechanism

According to **Hypothesis 2**, different concepts have different effects on exercises, and the relationship matrix $\mathbf{P} \in \mathbb{R}^{\mathcal{M} \times \mathcal{K}}$ of concepts and exercises is obtained by calculating the attention between exercises and concepts, where p_{mk} denotes the effect of concept c_k on exercise e_m , where $0 < p_{mk} < 1$ if $q_{mk} = 1$; otherwise, $q_{mk} = 0$, where q_{mk} denotes the elements contained in the relationship matrix \mathbf{Q} between exercises and concepts. \mathbf{P} denotes the quantitative interaction between exercises and concepts.

According to **Hypothesis 1**, there is an interaction between concepts. This interaction is quantitatively denoted as $\mathbf{T} \in \mathbb{R}^{\mathcal{K} \times \mathcal{K}}$, and t_{vk} ($0 < t_{vk} < 1$) denotes the influence of concept c_v on concept c_k , and for the k -th column of the \mathbf{T} , $\sum_{v=1}^{\mathcal{K}} t_{vk} = 1$. Therefore, \mathbf{T} denotes the quantitative interaction between concepts.

Attention Mechanism in the First Layer. In ACD, the parameter matrices $\mathbf{P}'' \in \mathbb{R}^{\mathcal{M} \times \mathcal{H}}$ and $\mathbf{T}'' \in \mathbb{R}^{\mathcal{K} \times \mathcal{H}}$ are defined to calculate the quantitative relationship matrices \mathbf{P} and \mathbf{T} , where \mathcal{H} is the hyperparameter, indicating the embedding dimension. The first layer of ACD, which calculates the learner's proficiency in some of the concepts. As shown in the top left of Fig. 2, the **Attention+Mask** module represents the Attention calculation and Mask operation, which is calculated as follows:

$$p_{mk} = \text{mask}(\text{sim}(\mathbf{p}''_m, \mathbf{t}''_k) \times \mathcal{G}, q_{mk}) = \text{sim}(\mathbf{p}''_m, \mathbf{t}''_k) \times \mathcal{G}, \quad (1)$$

$$\text{sim}(\mathbf{p}''_m, \mathbf{t}''_k) = \frac{\mathbf{p}''_m \cdot \mathbf{t}''_k^T}{\sqrt{\sum_{h=1}^{\mathcal{H}} p''_{mh}} \times \sqrt{\sum_{h=1}^{\mathcal{H}} t''_{kh}}}, \quad (2)$$

where p_{mk} is the element in $\mathbf{p}_m \in \mathbb{R}^{1 \times \mathcal{K}}$ that represents the similarity between exercise e_m and concept c_k . $\text{mask}(x, y) = -1e9$, if $y = 0$; otherwise, $\text{mask}(x, y) = x$. $\text{sim}(\mathbf{x}, \mathbf{y})$ is to calculate the similarity between the vectors \mathbf{x} and \mathbf{y} . \mathbf{p}''_m represents the initial embedding vector of exercise e_m . \mathbf{t}''_k represents the initial embedding vector of exercise c_k .

C_i denotes the set of concepts contained in $E_i^{(X)}$, as shown in Fig. 2, $E_i^{(X)} = \{e_2, e_3, e_4, e_6, e_8, e_9\}$. For example, if e_2 contains $\{c_1, c_3\}$, e_3 contains $\{c_2, c_5\}$, e_4 contains $\{c_3, c_5\}$, etc., then $C_i = \{c_1, c_2, c_3, c_5\}$. Without considering the interaction between concepts, the proficiency of learner l_i in C_i is denoted as \mathbf{a}_i^* . Thus, $|\mathbf{a}_i^*| = |C_i|$. \mathbf{a}_i^* is calculated as follows:

$$a_{ik}^* = \text{softmax}(p_{mk}) \times x_{im} = \sum_{e_m \in E_i^{(X)}} p_{mk}^{(i)} x_{im}, \quad (3)$$

where $P_{mk}^{(i)} = \frac{e^{p_{mk}}}{\sum_{e_v \in E_i^{(X)}} e^{p_{vk}}}$, x_{im} denotes the score of learner l_i on exercise e_m .

a_{ik}^* is an element of \mathbf{a}_i^* and denotes the proficiency of learner l_i on c_k without considering the interaction between concepts. $p_{mk}^{(i)}$ is the normalized weight of x_{ij} on c_k . p_{mk} and p_{vk} are elements of matrix \mathbf{P} .

Attention Mechanism in the Second Layer. Suppose, $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ and $C_i = \{c_1, c_2, c_3, c_5\}$, then $E_i^{(X)}$ does not involve concept $\{c_4, c_6\}$. If interactions between concepts are not considered, then l_i 's proficiency in $\{c_4, c_6\}$ is not diagnosable. Thus, the second layer of the ACD considers the interaction between concepts to diagnose learners' proficiency on all concepts. After adding the quantitative interaction matrix \mathbf{T} between concepts, the proficiency of learner l_i on all concepts was denoted as \mathbf{a}_i , calculated as follows:

$$t_{uk}^{(i)} = \text{softmax}(t_{uk}) = \frac{e^{t_{uk}}}{\sum_{c_v \in C_i} e^{t_{vk}}} \quad t_{uk} = \text{sim}(\mathbf{t}''_u, \mathbf{t}''_k) \times \mathcal{G}, \quad (4)$$

$$a_{ik} = \sum_{c_u \in C_i} t_{uk}^{(i)} a_{iu}, \quad (5)$$

where t_{uk} is an element in \mathbf{T} that represents the similarity of concept c_u to concept c_k . a_{ik} is the element in \mathbf{a}_i that represents the proficiency of learner l_i in c_k .

Attention Mechanism in the Third Layer. Based on learner l_i 's cognitive state \mathbf{a}_i , \mathbf{P}'' and \mathbf{T}'' , as shown in Fig. 2, learner l_i 's predicted score in $E_i^{(Y)} = \{e_1, e_5, e_7\}$ is denoted as $\hat{\mathbf{y}}'_i$, calculated as follows:

$$\hat{y}'_{im} = \sum_{k=1}^{\mathcal{K}} p_{mk}^{(m)} a_{ik} \quad p_{mk}^{(m)} = \text{softmax}(p_{mk}) = \frac{e^{p_{mk}}}{\sum_{v=1}^{\mathcal{K}} e^{p_{mv}}}, \quad (6)$$

where \hat{y}'_{im} is the element in $\hat{\mathbf{y}}'_i$ that represents learner l_i 's score on exercise e_m , $p_{mk}^{(m)}$ is the normalized weight of cognitive state a_{ik} relative to e_m .

However, even if the learner is proficient in all the concepts contained in the exercise, he/she may still answer the exercise incorrectly. In another case, even if the learner is not proficient in any of the concepts included in the exercise, he/she can still answer the exercise correctly. The former is called **slip**, and the latter is called **guess**. CDGK verified the quantitative benefits of slip and guess by ablation testing [23]. In this study, guess are denoted as $\mathbf{s} = \{s_1, s_2, \dots, s_{\mathcal{M}}\}$, and s_m denotes the rate of learners' lapses on exercise e_m . In ACD, define $\mathbf{s}'' \in \mathbb{R}^{1 \times \mathcal{M}}$ and $\mathbf{g}'' \in \mathbb{R}^{1 \times \mathcal{M}}$ to compute \mathbf{s} and \mathbf{g} , respectively. Specifically:

$$s_m = \frac{1}{1 + \exp(-s_m'')} \quad g_m = \frac{1}{1 + \exp(-g_m'')}, \quad (7)$$

where s_m'' is an element in \mathbf{s}'' , when initialized, $s_m'' = \varphi$, $\varphi = -2$ is an empirical constant; g_m'' is an element of \mathbf{g}'' , when initialized, $g_m'' = \varphi$.

When slip and guess are added, l_i 's predicted score on Exercise $E_i^{(Y)}$ is expressed as $\hat{\mathbf{y}}_i$, calculated by the following equation, $\hat{Y}_{im} = (1 - s_m)\hat{y}'_{im} + g_m(1 - \hat{y}'_{im})$, and \hat{Y}_{im} is the element of $\hat{\mathbf{y}}_i$ that represents l_i 's score prediction on Exercise e_m , $\hat{Y}_i = \{\hat{Y}_{im} | e_m \in E_i^{(Y)}\}$.

Finally, the *Loss* of the ACD model is calculated, as shown in Fig. 2, $\hat{\mathbf{y}}_i = \{y_{i1}, y_{i5}, y_{i7}\}$. The *Loss* of the model for learner l_i is defined as:

$$\text{Loss} = \frac{1}{\alpha \mathcal{N}} \sum_{\substack{y_{im} \in Y_i \\ \hat{y}_{im} \in \hat{Y}_i}} [-y_{im} \log(\hat{y}_{im}) + (1 - y_{im}) \log(1 - \hat{y}_{im})]. \quad (8)$$

The model parameters $\{\mathbf{P}''', \mathbf{T}''', \mathbf{s}''', \mathbf{g}'''\}$ are then updated by back propagation, using the Adam algorithm [12]. The optimization objective of the model is to minimize the *Loss*, which can be achieved by finite forward and backward propagation. Algorithm 1 shows the iterative training process of ACD.

4 Experiment

To validate the ACD model, a great number of experiments were conducted using five real datasets. In this section, the datasets, segmentation, evaluation criteria, baseline, experimental setup, and analysis of results are described.

Algorithm 1: The Iterative Process of ACD

Input: $\{R_i | 1 \leq i \leq N\}$: A record set of learners' answers; \mathbf{Q} : RelationshipsMatrix between exercises and concepts; α : The parameter of α -partition; \mathcal{G} : coefficients of the sim function; \mathcal{H} : Dimension of the embedding vector.

Output: \mathbf{A} : Cognitive state of learners; \hat{Y} : Predicted scores of learners.

```

1  $p''_{mh} \leftarrow \mathcal{N}(0, 1)$ ;  $t''_{kh} \leftarrow \mathcal{N}(0, 1)$ ; ( $1 \leq m \leq M$ ) ( $1 \leq k \leq K$ );
2  $s''_m \leftarrow -2$ ;  $g''_m \leftarrow -2$ ; ( $1 \leq m \leq M$ );
3  $\mathbf{A} \leftarrow \emptyset \in \mathbb{R}^{N \times K}$ ;  $\hat{Y}_i \leftarrow \emptyset$ ;  $Loss \leftarrow 0$ ;  $\sigma \leftarrow \text{sigmoid}$ ;  $\xi \leftarrow \text{mask}$ ;  $\gamma \leftarrow sim$ ;
4 for  $epoch \leftarrow 1$  to  $L$  do
5   for  $i \leftarrow 1$  to  $N$  do
6     Divide  $R_i$  in the training set into  $X_i$  and  $Y_i$  by  $\alpha$ -partition, there will produce  $\alpha X_i$  and  $Y_i$ , denoted as  $\{X_i^1, X_i^2, \dots, X_i^\alpha\}$  and  $\{Y_i^1, Y_i^2, \dots, Y_i^\alpha\}$  respectively;
7      $Loss_i \leftarrow 0$ ;
8     for  $z \leftarrow 1$  to  $\alpha$  do
9        $\hat{\mathbf{y}}_i \leftarrow \emptyset \in \mathbb{R}^{1 \times |E_i^{(Y)}|}$ ;
10      Denoted the element in  $X_i^z$  as  $x_{ij}$ , the element in  $\mathbf{a}_i$  as  $a_{ik}$ , the element in  $\hat{\mathbf{y}}_i$  as  $\hat{y}_{im}$ ;
11       $p_{mk} \leftarrow \xi(\gamma(p''_m, t''_k) \times \mathcal{G}, q_{mk})$ ;
12       $a_{ik}^* \leftarrow \sum_{e_m \in E_i^{(X)}} p_{mk}^{(i)} x_{im}$ ;  $P_{mk}^{(i)} \leftarrow \frac{\exp(p_{mk})}{\sum_{e_v \in E_i^{(X)}} \exp(p_{vk})}$ ;
13       $t_{uk} \leftarrow sim(\mathbf{t}''_u, \mathbf{t}''_k) \times \mathcal{G}$ ;
14       $t_{uk}^{(i)} \leftarrow softmax(t_{uk}) \leftarrow \frac{\exp(t_{uk})}{\sum_{c_v \in C_i} \exp(t_{vk})}$ ;
15       $a_{ik} \leftarrow \sum_{c_u \in C_i} t_{uk}^{(i)} a_{iu}$ ;
16       $p_{mk}^{(m)} \leftarrow softmax(p_{mk}) \leftarrow \frac{\exp(p_{mk})}{\sum_{v=1}^K \exp(p_{mv})}$ ;
17       $\hat{y}'_{im} \leftarrow \sum_{k=1}^K p_{mk}^{(m)} a_{ik}$ ;
18       $\lambda_m \leftarrow \sigma(\lambda''_m)$ ;  $s_m \leftarrow \sigma(s''_m)$ ;  $g_m \leftarrow \sigma(g''_m)$ ;
19       $Loss_i \leftarrow Loss_i - \sum_{\substack{y_{im} \in Y_i \\ \hat{y}_{im} \in \hat{Y}_i}} [y_{im} \log(\hat{y}_{im}) + (1 - y_{im}) \log(1 - \hat{y}_{im})]$ ;
20    end
21     $Loss \leftarrow Loss + Loss_i / \alpha$ ;
22    if  $epoch = L$  then
23      Add  $\mathbf{a}_i$  to set  $\mathbf{A}$ ;
24      Add  $\hat{Y}_i$  to set  $\hat{Y}$ ;
25    end
26  end
27   $Loss \leftarrow Loss / N$ ;
28  Update  $\mathbf{P}''$ ,  $\mathbf{T}''$ ,  $\mathbf{s}''$ ,  $\mathbf{g}''$ , and  $Loss$ ;
29 end
30 Return  $\mathbf{A}$  and  $\hat{Y}$ .

```

4.1 Datasets and Preprocessing

Datasets: This paper uses five real-world datasets, including a small-scale dataset FrcSub¹ and four large-scale datasets: ASSIST2009², ASSIST2017³, JunYi⁴ and MathEC⁵.

¹ <http://staff.ustc.edu.cn/~qiliuql/data/math2015.rar>.

² <https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010>.

³ <https://sites.google.com/view/assistmentsdatamining/data-mining-competition-2017>.

⁴ <https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198>.

⁵ <https://eedi.com/projects/neurips-education-challenge>.

Preprocessing: For all datasets, if exercises more than once by a learner, then only the first response was used. To ensure that each learner had enough response records for cognitive diagnosis, this paper selected learners with response records greater than 15 for all datasets. “#Learners” denotes the number of learners, “#Exercises” denotes the number of exercises, “#Concepts” denotes the number of concepts, “#Answered record” denotes the number of answer records, and “#Concepts interaction” denotes the number of concept interactions. Table 2 summarizes the basic statistics of these datasets.

Table 2. The descriptions of the datasets.

CDMs	FrcSub	ASSIST2009	ASSIST2017	JunYi	MathEC
#Learners	536	2380	1678	36591	118971
#Exercises	20	16804	2210	721	27613
#Concepts	8	110	101	721	388
#Concepts interaction	0	0	0	1918	387
#Answered exercises	10720	257585	351530	1550016	15867850

Division of Dataset: These datasets were divided according to 8: 2 depending on learners, where 80% of the fraction represents historical learners and is used to train ACD, where the other 20% of the data, represented as new learners. For each new learner, their 20% answer records were used to predict the scores on the remaining 80% of the exercises.

4.2 Baselines and Evaluation Metrics

To validate the performance of ACD, this paper uses two classical CDMs, MIRT [16] and DINA [9], and four state-of-the-art CDMs, NCDM [22], CDGK [23], RCD [8] and QRCDM [24] for comparison.

In this study, the exercises in the dataset used were all objective questions, so the scores of the exercises were either 0 or 1, and the predicted learners’ scores were between 0 and 1. To verify the accuracy of the ACD model in predicting learner performance, the classification metrics Prediction Accuracy (*ACC*) and Area Under a ROC Curve (*AUC*), and the regression metric Root Mean Square Error (*RMSE*) were chosen for evaluation.

4.3 Parameters Sensitivity Analysis and Setting

In this paper, all experiments were conducted using Python 3.8.0 and Pytorch 1.10.0, running on an Linux-3.10.0 platform, with a running memory capacity of 128 GB, and all experiments were accelerated by RTX 2080 Ti. To ensure fairness, all models were selected for optimal performance.

The hyperparameters of ACD model mainly contain: α (parameter in α -partition), batch size, training set ratio, \mathcal{H} (embedding dimension), \mathcal{G} (coefficient of sim function). The effect of these hyperparameters on ACD is analyzed on five datasets to verify the robustness of ACD. Figures 3, 4, 5, 6 and 7 show the trends of AUC and $RMSE$ when the ACD hyperparameters are varied.

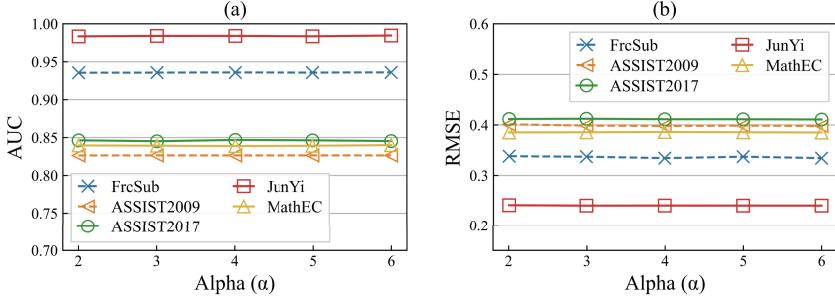


Fig. 3. The variation of the ACD performance with α .

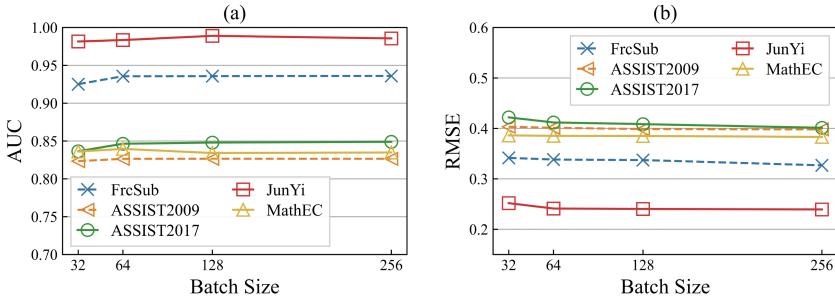


Fig. 4. The variation of the ACD performance with batch size.

From Figs. 3, 4, 5, 6 and 7, it is evident that ACD is not sensitive to parameters, and changes are basically stable. Considering the need to conserve computing resources, parameters are set as follows: α is set to 2, \mathcal{H} is set to 64, the batch size is set to 64, the training set ratio is set to 0.8, and \mathcal{G} is set to 5.

4.4 Experimental Results

Table 3 and Table 4 show the experiments of the ACD and baseline models on the five datasets, respectively, and all models are selected for optimal performance.

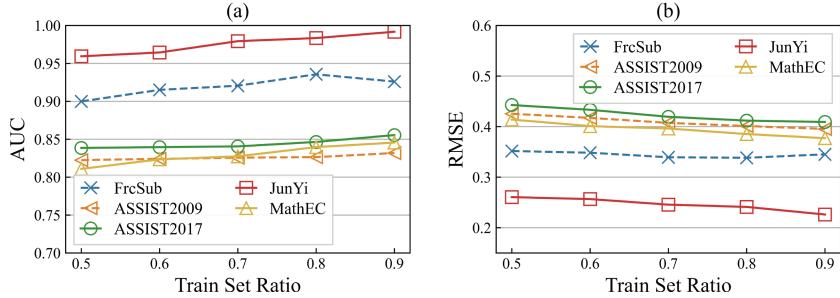


Fig. 5. The variation of the ACD performance with train test ratio.

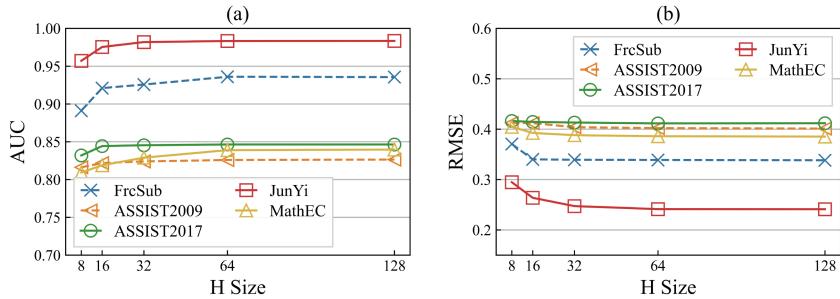


Fig. 6. The variation of the ACD performance with \mathcal{H} .

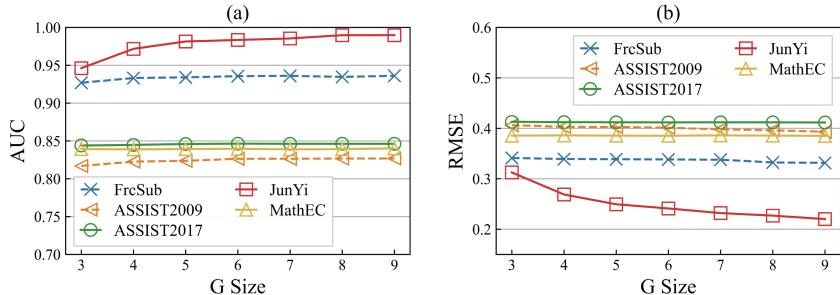


Fig. 7. The variation of the ACD performance with \mathcal{G} .

As shown in Table 3 and Table 4, the model of ACD is generally significantly higher than the baseline models. The baseline models are less effective in predicting new learners. In addition, in the experimental Baselines performance, QRCDM, RCD, CDGK, and NCDM outperformed DINA and MIRT, indicating that deep neural networks are better at capturing the complex relationships of exercises and concepts. In addition, the results of QRCDM are better than other baseline models, which may be related to the fact that QRCDM didn't use the learner's id as the embedding vector when constructing.

Table 3. Performance of the prediction scores of CDMs on partial datasets. Bold denotes the best results.

CDMs	FrSub			ASSIST2009			ASSIST2017		
	ACC ↑	AUC ↑	RMSE ↓	ACC ↑	AUC ↑	RMSE ↓	ACC ↑	AUC ↑	RMSE ↓
MIRT	0.5476	0.5624	0.6234	0.5479	0.5410	0.6489	0.6184	0.6402	0.5235
DINA	0.5023	0.6343	0.6244	0.5118	0.5923	0.5645	0.5130	0.6014	0.5649
NCDM	0.6104	0.6529	0.4985	0.6710	0.6755	0.4578	0.6535	0.7014	0.4660
CDGK	0.6086	0.6424	0.4836	0.6744	0.6646	0.4586	0.6460	0.6890	0.4698
RCD	0.5312	0.5672	0.4987	0.6611	0.5876	0.4682	0.5787	0.5696	0.4943
QRCDM	0.8335	0.9137	0.3586	0.7517	0.8166	0.4203	0.7486	0.8267	0.4131
ACD	0.8528	0.9356	0.3382	0.7645	0.8264	0.4011	0.7677	0.8464	0.4118

Table 4. Performance of the prediction scores of CDMs on another subset of datasets. Bold denotes the best results.

CDMs	JunYi			MathEC		
	ACC ↑	AUC ↑	RMSE ↓	ACC ↑	AUC ↑	RMSE ↓
MIRT	0.6978	0.5435	0.5617	0.5600	0.5561	0.6185
DINA	0.5001	0.6338	0.5097	0.5024	0.5511	0.6013
NCDM	0.7939	0.7849	0.3799	0.6590	0.6644	0.4664
CDGK	0.7912	0.7770	0.3822	0.6639	0.6694	0.4603
RCD	0.7647	0.5812	0.4221	0.6442	0.5955	0.4739
QRCDM	0.8894	0.9362	0.2861	0.7812	0.8148	0.3859
ACD	0.9535	0.9834	0.2411	0.7931	0.8396	0.3852

5 Conclusion

This paper proposes a cognitive diagnostic model ACD based on a three-layer attentional mechanism neural network. The outputs and parameters of each layer of the network are real numbers between 0 and 1, all of which are interpretable. Meanwhile, the ACD solves the cold start problem for new learners and has good interpretability.

In future work, we will explore how to build on existing research for practice recommendations and academic alerts to improve learner motivations and learning efficiencies.

Acknowledgement. This work is partly supported by the National Natural Science Foundation of China under Grant No. 61977044 and 62077035; the Ministry of Education’s Cooperative Education Project Grant No. 202102591018; This work is partly supported by the Key Laboratory Funds of the Ministry of Culture and Tourism under grant No. 2022-13.

References

1. Castro-Schez, J.J., Glez-Morcillo, C., Albusac, J., Vallejo, D.: An intelligent tutoring system for supporting active learning: a case study on predictive parsing learning. *Inf. Sci.* **544**, 446–468 (2021). <https://doi.org/10.1016/j.ins.2020.08.079>
2. Chaudhari, S., Mithal, V., Polatkan, G., Ramanath, R.: An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol.* **12**(5), 32 (2021). <https://doi.org/10.1145/3465055>
3. Cheng, S., et al.: DIRT: deep learning enhanced item response theory for cognitive diagnosis. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), pp. 2397–2400. ACM (2019). <https://doi.org/10.1145/3357384.3358070>
4. Chung, J.Y., Lee, S.: Dropout early warning systems for high school students using machine learning. *Child. Youth Serv. Rev.* **96**, 346–353 (2019). <https://doi.org/10.1016/j.childyouth.2018.11.030>
5. Ellis, H.C.: The Transfer of Learning. Macmillan, New York (1965)
6. Embretson, S.E., Reise, S.P.: Item Response Theory. Psychology Press, New York (2013)
7. Gao, L., Zhao, Z., Li, C., Zhao, J., Zeng, Q.: Deep cognitive diagnosis model for predicting students' performance. *Future Gener. Comput. Syst.* **126**, 252–262 (2022). <https://doi.org/10.1016/j.future.2021.08.019>
8. Gao, W., et al.: RCD: relation map driven cognitive diagnosis for intelligent education systems. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 501–510. ACM (2021). <https://doi.org/10.1145/3404835.3462932>
9. de La Jimmy, T.: Dina model and parameter estimation: a didactic. *J. Educ. Behav. Stat.* **34**(1), 115–130 (2009). <https://doi.org/10.3102/1076998607309474>
10. de La Jimmy, T.: The generalized DINA model framework. *Psychometrika* **76**(2), 179–199 (2011). <https://doi.org/10.1007/s11336-011-9207-7>
11. Kamii, C.: The equilibration of cognitive structures: the central problem of intellectual development. *Am. J. Educ.* **94**(4), 574–577 (1986)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. Liu, Q., et al.: Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Trans. Intell. Syst. Technol.* **9**(4), 48:1–48:26 (2018). <https://doi.org/10.1145/3168361>
14. Lord, F.M.: Applications of Item Response Theory to Practical Testing Problems. Routledge, London (2012)
15. Pinar, W.F., Reynolds, W.M., Taubman, P.M., Slattery, P.: Understanding Curriculum: An Introduction to the Study of Historical and Contemporary Curriculum Discourses. Peter Lang, New York (1995)
16. Reckase, M.D.: Multidimensional Item Response Theory Models. Springer, New York (2009). <https://doi.org/10.1007/978-0-387-89976-3>
17. Rosenbaum, P.R.: Testing the conditional independence and monotonicity assumptions of item response theory. *Psychometrika* **49**(3), 425–435 (1984). <https://doi.org/10.1007/BF02306030>
18. Templin, J.L., Henson, R.A.: Measurement of psychological disorders using cognitive diagnosis models. *Psychol. Methods* **11**(3), 287 (2006). <https://doi.org/10.1037/1082-989X.11.3.287>

19. Tong, S., et al.: Item response ranking for cognitive diagnosis. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1750–1756. ijcai.org (2021). <https://doi.org/10.24963/ijcai.2021/241>
20. Treagust, D.F., Chandrasegaran, A.L.: Assessment of students' conceptual understandings in science: the Taiwan national science concept learning study in an international perspective. *Int. J. Sci. Educ.* **29**(4), 391–403 (2007). <https://doi.org/10.1080/09500690601072774>
21. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
22. Wang, F., et al.: Neural cognitive diagnosis for intelligent education systems. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI), pp. 6153–6161. AAAI Press (2020). <https://doi.org/10.1609/aaai.v34i04.6080>
23. Wang, X., Huang, C., Cai, J., Chen, L.: Using knowledge concept aggregation towards accurate cognitive diagnosis. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), pp. 2010–2019. ACM (2021). <https://doi.org/10.1145/3459637.3482311>
24. Yang, H., et al.: A novel quantitative relationship neural network for explainable cognitive diagnosis model. *Knowl.-Based Syst.* **250**, 109156 (2022). <https://doi.org/10.1016/j.knosys.2022.109156>
25. Zhang, H., Huang, T., Lv, Z., Liu, S., Zhou, Z.: MCRS: a course recommendation system for MOOCs. *Multimed. Tools Appl.* **77**, 7051–7069 (2018). <https://doi.org/10.1007/s11042-017-4620-2>



Data Augmentation for Knowledge Tracing Based on Variational AutoEncoder and Efficient Network Reusing

Hui Zhao and Jun Sun^(✉)

Wangxuan Institute of Computer Technology, Peking University, Beijing, China
hui.zhao@stu.pku.edu.cn, sunjun@pku.edu.cn

Abstract. Knowledge tracking (KT) is a task that predicting the degree of students' knowledge mastery through their learning interaction records. Although existing works improve predictive capability with well-designed neural network models or hypothetical learning mechanisms, the predictive performance is compromised in the scenarios of quantity limited interaction data. In this paper, we utilize Variational AutoEncoder (VAE) and pre-trained network to generate question-answer sequence data pairs related to the original interaction data, which can improve the performance of the model when added to the training set even in the case of data scarcity. Specifically, the steps of the data augmentation method for KT we proposed are as follows: 1) Question sequence generation. Generate latent question sequences that are similar to the real interaction question sequences from the pre-designed VAE model. 2) Answer sequence generation. Put the generated data into the pre-trained KT model to get reliable answer label sequences that correspond to latent question sequences. 3) Samples generation and training. Combine the two types of generated sequences as new samples for KT task training. We apply the data augmentation method on four classic datasets and demonstrate its effectiveness by reaching the state-of-the-art performance with an average AUC index improvement of 2.41%. We also verify the method on artificially random extracted data, and with only 20% of the data, it even achieves similar results compared with other methods using 100% of the data.

Keywords: Knowledge tracking · Sample generation · Data augmentation · Variational autoencoder

1 Introduction

Online education is a necessary supplement to school learning [25] that provides students with access to a wide range of resources, flexible and personalized learning ways, and a convenience of learning anytime and anywhere [30–32]. For online education system, the basis of learning strategy is to understand the knowledge state of students [28], because it related to learning path planning [27], personalized exercise recommendation [24], and other downstream tasks [9, 10]. Knowledge tracking (KT) is a task of predicting students' knowledge state and mastery based on their existing answered questions. As

shown in Fig. 1(a), the goal of KT is to predict the probability that the student can answer the certain question correctly when the former questions (include their answered state) and the corresponding knowledge are given. For instance, in Fig. 1(a), one student (S_1) answered three exercises (e_1, e_2, e_3) where e_1 and e_3 were correct but e_2 was incorrect. KT's target is to predict the probability of S_1 gets e_4 right.

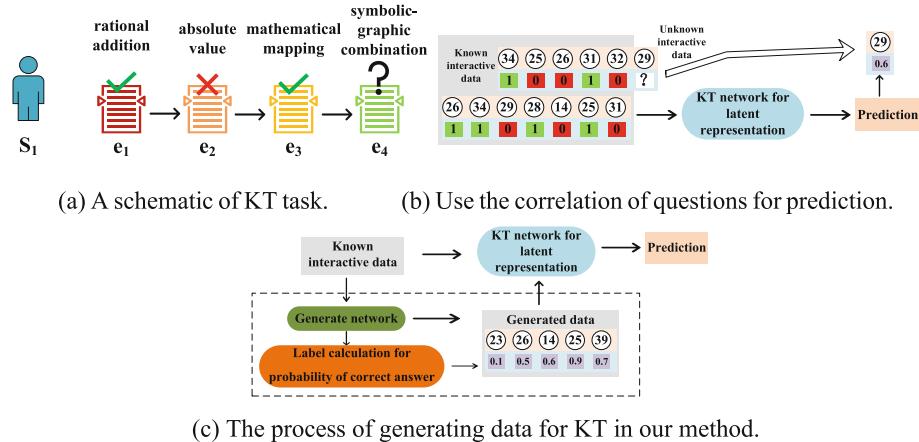


Fig. 1. (a) The goal of KT is to predict the probability that the student can correctly answer the next question. (b) Existing research infers the probability that a certain student gets the answer right from other students' interaction records. The numbers in the circle represent the index of questions, and the numbers in the box represent the answered state or probability of getting it right. (c) The process of our data augmentation method.

In order to improve the accuracy of KT prediction, previous works shifted from logistic regression model and Bayesian method to deep learning. Deep Knowledge Tracing (DKT) [2] is the first deep neural network proposed to solve KT with Recurrent Neural Network (RNN) to model the students' knowledge state by using hidden vectors. On the basis of DKT, the subsequent researches mainly focus on designing appropriate neural network models [3, 4, 13] and hypothesizing the learning mechanisms of students [8, 11, 17, 21]. Scholars added learning factors for neural networks and took the awareness of learning process into account, such as memory mechanism with networks [3], content of exercises [5], semantic representations [6], context [7] and forgetting mechanism [8].

Although the well-designed networks and introduced learning mechanisms can effectively improve KT performance, they do not adequately address the problem of data scarcity. Due to the various number of questions and the disparate logical relationships to the questions in different datasets, it is difficult for the KT model to migrate between datasets. Difference of subjects or knowledge labels will lead to the failure of the model, which denotes that it is important to train the KT model from limited amount of realistic interaction data. To make full use of the data, as shown in Fig. 1(b), existing researches infer the probability that a certain student gets the answer right from other students' interaction records. This method can capture the correlation between questions and reduce the dependence of the student's interaction data. It relies on the interaction records of

other students, but the poor portability of records in real interactions are still insufficient nevertheless. Following the existing research method and mechanism [1], in this paper, we generate potential interactive representation vectors from the original data. The generated data has been verified to be helpful for improving the performance of the KT model, and when the original data is scarce, adding the potential interactive representation data could greatly improve the accuracy of model's prediction.

Formally, we regard one student's learning interaction record as a point in a multi-dimensional vector space, and the generated data is equivalent to a set of points closest to other original points in the vector space. When there are few real points in the vector space to represent students' knowledge state, the generated points can compensate for the deficiency of data.

In order to produce effective presentation data, we propose a generation model based on Variational AutoEncoder (VAE), which can generate data different from the original data but with training value. Since the interactive data is composed of a pair sequences of knowledge labels and answering states, the specific generation steps are as follows. Firstly, we input the interactive data (knowledge label sequences corresponding to the questions) into the designed VAE model to obtain vectors similar with original known data. Secondly, we treat these vectors as latent representation of the known data, and input them into the trained KT model to calculate probability labels of answering correctly corresponding to each question. Thirdly, the latent representation vectors and the labels of probability sequences are combined as generated data, which can be used for training and improving the performance of KT. The entire process of generating data for KT is shown in Fig. 1(c).

Our contributions are summarized as follows:

- In this article, unlike previous studies, we improve the prediction accuracy of KT from the perspective of data augmentation, which effectively suppresses the impact of data scarcity in the KT task.
- We generate latent data representations from the designed VAE model and the pre-trained network. As far as we know, it is the first time using this method for KT, and we design an efficient model suitable for the generated data.
- Our proposed Data Augmentation KT (DAKT) method achieves state-of-the-art performance on four classical datasets, with an average improvement of 2.41%.

2 Related Work

2.1 Knowledge Tracing Task

KT is a predictive task that utilizes students' past answer records to estimate the probability of a certain student correctly answering the next exercise. The prediction results can be considered as a model of students' knowledge and abilities, making KT an upstream task for other educational application tasks.

As a data mining task in the field of education, KT has been the subject of extensive research even before the widespread use of deep learning [18, 20]. Item Response Theory (IRT), Bayesian Knowledge Tracing (BKT), and logistic regression models have approached the KT task from the perspectives of data statistics and parameter modeling [10, 22, 23]. DKT utilizes RNNs for the KT task, marking the beginning of a new era

of methods in the deep learning age. It not only reduces reliance on expert experience but also significantly improves the predictive performance of the model [2]. Building on the foundation of DKT, structures such as Dynamic Key-Value Memory Networks (DKVMN) [3], Long Short-Term Memory (LSTM) networks, and self-attention mechanisms [4] have been applied to the architecture of deep learning networks for KT tasks [16, 19].

Under the research framework of deep learning, previous studies have promoted KT tasks from the perspectives of designing or introducing network models and creating assumptions for learning processes [39]. These two methods extract the potential information of the original data through the layers in the neural network. However, if the network is not designed properly or the network not designed for KT is used, it may be difficult to take advantage of the available information. In this paper, we add effectiveness proven components from previous works to our network, but more importantly, we extend the representation of input data by generative models that generate data for training.

2.2 Generative Models for Training

Generative models are a class of machine learning models that synthesize new data instances from a learned distribution. These models are trained on a dataset to capture its underlying patterns and variability, enabling them to generate new samples that are statistically similar to the training data. Generative models, such as Generative Adversarial Networks (GANs), Auto-Regressive model (AR), Normalizing Flow (Flow), Diffusion probabilistic model (Diffusion) [40] and Variational Autoencoders (VAEs) [33], have seen rapid development and widespread adoption in recent years [34, 36], driven by their ability to generate high-quality, realistic data. GANs, for instance, pit a generator network against a discriminator network in a min-max game, resulting in increasingly realistic generated samples. These generated data can be used to train other machine learning models, enhancing their performance and robustness, particularly in scenarios where labeled data is scarce or expensive to obtain [37]. The use of generative models and synthetic data is an active area of research and holds great promise for advancing the field of machine learning.

However, using generated data for training has its limitations. The quality and diversity of the generated data may not always be sufficient for effective model training, and care must be taken to ensure that the generated data does not introduce biases or other issues. Additionally, adding too much analogous generated data for training may lead to overfitting. In this paper, we verify the effectiveness of the generated data, using a simple linear fully connected network as the basic component of the generative model to avoid bias, and add regularization methods to avoid drift of fitting.

2.3 VAE

VAEs are a type of generative model that learn a probability distribution over the input data. VAEs learn a probabilistic encoding of the data and use this representation to generate new instances [35]. They consist of an encoder, which maps inputs to a latent space, and a decoder, which generates outputs from the latent space. VAEs are trained

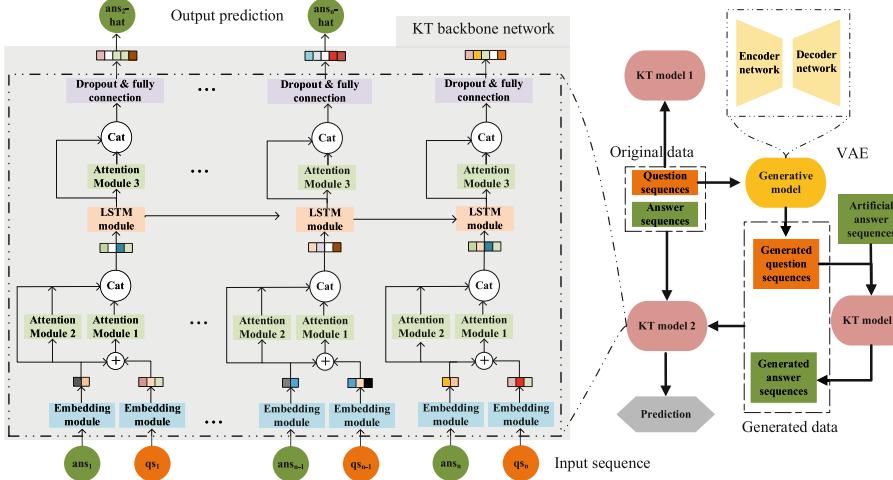


Fig. 2. The process of data augmentation and the network structure of DAKT. We generate question sequences as latent representations from VAE, then get generated answer sequences from the KT network with artificial answer sequences corresponding to the generated question sequences. The combination of the two types of generated sequences will be input into another untrained KT network, together with the original data. The two KT models are structurally identical, but with different parameters. We set the three attention modules identically, and more details of the network can be found in our paper.

to maximize the evidence lower bound (ELBO), which encourages the model to generate realistic samples and learn meaningful representations. The latent space allows for interpolation and generation of new data points, making VAEs useful for tasks like generative modeling, unsupervised learning, and representation learning.

Considering the flexibility of data generation and the trainability of the model, we adopt VAE to generate data which compensates for the deficiency of representation on the original data.

3 The Data Augmentation KT Model

3.1 Overview and Formulation

We present our Data Augmentation KT (DAKT) method in Fig. 2. As shown in the figure, we first design a VAE generative model to generate data similar to the question sequences in the original interactive data. Then we utilize the original data to train KT model 1 and the generated question sequences data from VAE will be input into the KT model 1 to produce the probability sequences of answering correctly (call it answer sequences for simplicity). The generated question sequences and the generated answer sequences combine to create generated data, which, in terms of structure and similarity, is identical to the original data. Different from previous works, DAKT feeds not only original data but also generated data into the KT model 2 and get the predicted probabilities by mapping the output of the last layer. In particular, KT model 1 and KT model 2 are structurally identical, but with different parameters.

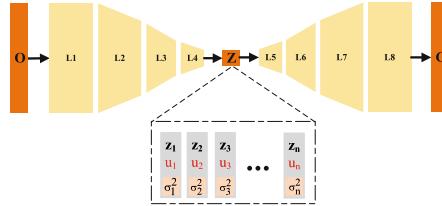


Fig. 3. Schematic diagram of the generative VAE model.

We define a student's collection of learning data as $\mathbf{Y}_n = \{y_1, y_2, \dots, y_n\}$, where y_t is the t -th interaction record. For each y_t , $y_t = (qst, ans_t)$, where qst is the number label of knowledge corresponding to a certain exercise and ans_t stands for the answered state of the t -th exercise. Formally, we describe the KT task in terms of conditional probability, i.e. $p(ans_m = 1 | qst_m, Y_{m-1})$. In the model, qs and ans will be used as input to the neural network. These two pieces of data appear in pairs, where the former, represented by a numerical label, denotes the knowledge concept corresponding to the exercise, and the latter, indicated by 0 or 1, represents whether the exercise was answered correctly. Each y has a given length L . If the length of one interactive data exceeds L , it will be split into multiple ones. If the length is shorter than L , the rest will be filled with paddings.

3.2 Generative Model

In order to reduce the impact of model bias on the generated data, we construct the generative VAE model with linear layers. As a generative probabilistic model that learns the latent representation of original data, following previous works [33, 35], our VAE model consists of encoder module and decoder module. As shown in Fig. 3, the encoder part includes four linear layers that map the original data to the latent space, capturing its statistical properties. The decoder part of the VAE also includes four linear layers that are symmetric with the encoder, which map the latent representations back to the data space, generating reconstructions of the original data. Given the original learning interaction data O , we expect the encoder network to learn low-dimensional features of the data as Z , which can be reconstructed to vectors as G that are similar to O by the decoder.

In the VAE network, we aim to generate sample points that are similar to the original data. However, the sample points in the original data consist of question indexes and corresponding answer status labels. The latter can be predicted by the network but cannot be generated directly, otherwise, the generated data would not be meaningful. The data generated by VAE consists of sequences of question indexes, simulating the order in which students answer questions (exercises).

3.3 Data Generation

As previously mentioned, the original data is composed of sequences of question indexes and sequences of answer labels, and the two types of sequences are in one-to-one correspondence. The generated sequences of questions can be treated as extensions of the original data in latent space, but the generated method does not make sense for the

answer labels. Thus, in our method, we use VAE to generate the question sequences and input the sequences into pre-trained KT network to produce reliable answer labels. As shown in Fig. 2, we first train the KT model 1 network with the original data and fix parameters for use next. Then we create artificial answer sequences, which will be input into KT 1 along with the generated question sequences from VAE. Since the KT 1 network has been pre-trained, the input of the two types of sequences is equivalent to fine-tuning the parameters of the KT 1 model. For convenience and with limited influence on fine-tuning, we set all of the generated questions to be answered correctly, in other words, we set artificial answer sequences to ones. In this way, reliable answer sequences corresponding to question sequences which produced from VAE can be generated. At last, we combine the generated question sequences and generated answer sequences as generated data, which will be mixed together with original data as the input of KT 2 network. It's important to emphasize that KT 1 and KT 2 are structurally identical, but different in parameters because of the data they're training on.

3.4 KT Backbone Network

As shown in Fig. 2, we design the KT backbone network which is composed of embedding modules, attention modules, long short-term memory (LSTM) modules, dropout layers and full connection layers.

Data Processing Procedure. In Fig. 2, the combination of different colored squares represents the processed vectors. The embedding module maps the sparse interaction data to the space with relatively lower dimensionality. The question sequences and the answer sequences are processed by embedding modules, and then they will be input into attention module 1. The next concatenation is a simple joining operation which combines the answer embeddings with the two attention modules' outputs. Those three streams of information are aggregated and fed into the LSTM modules. Output vectors from LSTM modules will be input into the attention module 3, which will be concatenated with the output of attention module 3. The concatenation of the two vectors is connected to the dropout and fully connection layers. Two small tricks are used in the KT backbone network. One is to use the short connection strategy to retain the original information and the other is to use the dropout method to avoid drift of fitting caused by the generated data.

Table 1. An overview of four benchmark datasets

Dataset	Students	Knowledge concepts	Interactions	Average records
ASSISTments2009	4151	110	325,637	2500
Statics2011	333	1,223	189,297	230
ASSISTments2015	19,840	100	683,801	11904
ASSISTments2017	1,709	102	942,816	1180

LSTM Module and Attention Module. LSTM module recurrently use the information of current vectors and precious information to predict current data. We connect LSTM module and multiple attention modules to increase the simulation ability of the network. Attention module transforms the raw vectors into the attentive vectors. We introduce the attention mechanism [14] due to the association of knowledge, that is to say, whether a student can answer one question correctly is related to whether the student can answer other questions correctly.

By the attention mechanism, attention module pays attention to those important and basic knowledge skills correspond to questions or presentations in latent space. And at the same time, attention module retains the original information as much as possible by the operation of concatenation. As shown in Fig. 2, we introduce three attention modules as long-term attention which enables the original information to be preserved in the form of convolution.

4 Experiments

4.1 Datasets and Statement

We have conducted validations on four classic datasets, and the specific attributes of these datasets are shown in Table 1. Here, it is necessary to explain the characteristics of the datasets. KT datasets contain the sequence numbers of the knowledge concepts corresponding to exercises and the labels of response status. Different exercises may correspond to the same knowledge concept. The response status labels are binary, with only two options, 0 and 1, indicating a wrong or correct answer, respectively.

4.2 Implementation Details

VAE Network. Across the four datasets, we used the same hyperparameters for the VAE network. We set the length of the interactive data L as 200, and the number of nodes in our four linear network layers of the encoder is determined by L . The input/output dimensions of the four linear layers are (L, L) , $(L, L/1.5)$, $(L/1.5, L/2)$, $(L/2, L/4)$ respectively. For the decoder, input/output dimensions are symmetric, i.e. $(L/4, L/2)$, $(L/2, L/1.5)$, $(L/1.5, L)$, (L, L) . During the training, we use the Adam optimization method and set learning rate = 0.001.

KT Network. Due to the differences in attributes and sizes of the datasets, we set different hyperparameters for each dataset in the KT backbone network. In all the experiments below, the embeddings of the questions on the four datasets are set to 128, 256, 36 and 128, respectively. The embeddings of the answers on the four datasets are set to 64, 64, 36 and 64, respectively. The default number of hidden dimensions are 128, 96, 64 and 128, respectively. We set the patience strategy value of training as 25 and set learning rate = 0.001. We set the dimension of all attention modules as 80. The values in the other modules can be deduced by the structure of the network. In order to prevent fitting deviation caused by the excessive generated data, we set the values of dropout to 0.5, 0.9, 0.1, and 0.5, respectively, depending on the amount of the interactive sequences.

Table 2. The comparison of several methods on the performance of AUC. The results achieved by our method are highlighted with bold fonts. To make an intuitive comparison, we list the average performance across the four datasets for each method. On every dataset, DAKT achieves the best performance among all methods.

Method	Dataset				
	ASSISTments2009	Statics2011	ASSISTments2015	ASSISTments2017	Average
BKT+ [10]	~0.69	~0.75	\	\	\
DKT [2]	0.8170 ± 0.0043	0.8233 ± 0.0039	0.7310 ± 0.0018	0.7263 ± 0.0054	~0.7744
DKT+ [11]	0.8024 ± 0.0045	0.8301 ± 0.0039	0.7313 ± 0.0018	0.7124 ± 0.0041	~0.7691
AKT [7]	0.8169 ± 0.0045	0.8265 ± 0.0049	0.7828 ± 0.0019	0.7282 ± 0.0037	~0.7886
SAKT [4]	0.7520 ± 0.0040	0.8029 ± 0.0032	0.7212 ± 0.0020	0.6569 ± 0.0027	~0.7333
DKVMN [3]	0.8093 ± 0.0044	0.8195 ± 0.0041	0.7276 ± 0.0017	0.7073 ± 0.0044	~0.7659
ATKT [12]	0.8244 ± 0.0032	0.8325 ± 0.0043	0.8045 ± 0.0097	0.7297 ± 0.0051	~0.7978
DAKT	0.8396 ± 0.0029	0.8401 ± 0.0035	0.8669 ± 0.0039	0.7410 ± 0.0036	~0.8219

4.3 Comparison with the State-of-the-Arts

Table 2 shows the comparison results of our proposed method (DAKT) with other state-of-the-art methods. Roughly speaking, our method performs the best on all four datasets.

Due to the different amounts of data in the four datasets, the improvement of our method varies across the datasets. Statics2011 has only 230 interactive sequences per training set on average, while in ASSISTments2015, the number is 11904. The small number of interactive sequences means that the data augmentation strategy is not effective for the dataset, and Statics2011 only achieves an improvement of 0.76% compared with ATKT. In contrast, the interactive sequences of ASSISTments2009 and ASSISTments2017 are 2500 and 1180, and their improvements are 1.52% and 1.13%, which compared with ATKT.

4.4 Test in Data Scarcity Scenarios

For online learning, it is difficult to obtain all the interactive data before KT task. Moreover, when the mapping of knowledge skills to the corresponding questions changes, the network needs to be retrained, which means it is important to perform KT task on existing quantity limited data. By using part of original data for training, we have demonstrated our method’s ability to generate efficient representations through data augmentation. On the four datasets, we test the performance of the training set at only 20%, 50%, and 80% remains, and generate half of the original data volume to compensate for the loss of the cutdown. As shown in Table 3, using only 20% of the data, DAKT achieves similar results compared to other methods using full data. In contrast, we conduct a similar experiment on original data without data augmentation. The results show that the data

augmentation strategy could effectively reduce the impact of data scarcity because it can generate potential information representations to offset the loss of data.

Table 3. Performance of the model in data scarcity scenarios with or without data enhancement. In the table, DA stands for data augmentation.

With DA	The percentage of using original data			
Dataset	20%	50%	80%	100%
ASSISTments2009	79.17%	82.17%	83.11%	83.96%
Statics2011	80.21%	82.70%	83.25%	84.01%
ASSISTments2015	82.15%	84.78%	86.08%	86.69%
ASSISTments2017	70.36%	72.73%	73.60%	74.10%
Without DA	The percentage of using original data			
Dataset	20%	50%	80%	100%
ASSISTments2009	74.64%	79.28%	81.69%	83.17%
Statics2011	77.25%	81.21%	82.19%	82.86%
ASSISTments2015	78.90%	83.42%	84.87%	85.53%
ASSISTments2017	67.21%	70.29%	72.58%	72.93%

Table 4. Effect of different yields of generation data on model performance. The generation rate is relative to the original data.

Dataset	Generation rate				
	0.05	0.1	0.2	0.5	1.0
ASSISTments2009	83.79%	83.84%	83.54%	83.96%	83.82%
Statics2011	83.98%	83.92%	84.01%	83.79%	83.83%
ASSISTments2015	86.69%	86.55%	86.52%	86.60%	86.59%
ASSISTments2017	74.01%	74.10%	73.93%	73.88%	74.08%

4.5 Generation Quantity Experiment

In this part, we study the effect of different generation quantities on performance of the KT model. From VAE and pre-trained model, we generate 5%, 10%, 20%, 50% and 100% of the original data and input them into the KT model together with the original data. As is shown in Table 4, there is no essential difference between 5% generated data and 100% generated data for the performance of DAKT. There are two main reasons to

explain this phenomenon. First, the VAE model has well-learned the data representations in latent space, and more generated data is just another form of the representations, which has a slight gain for the performance of KT. Second, the gains of performance is more from the KT network than the data augmentation strategy. It should be emphasized that too much generated data may cause more serious drift of fitting, which will affect the performance of the model. Data augmentation is important but with a moderate amount. However, less data means less training time, which helps improve the efficiency of the model. More discussion will be presented in the ablation experiments.

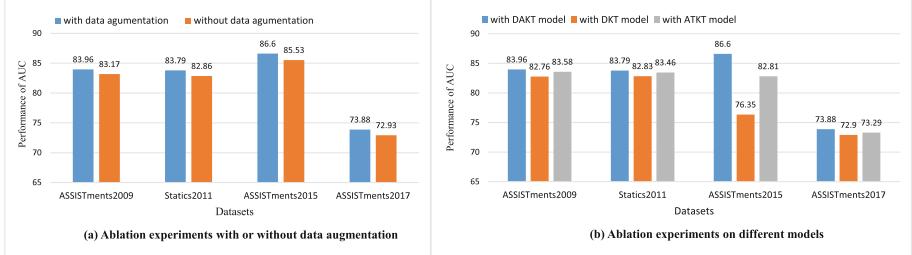


Fig. 4. Diagrams of main ablation experiments on data comparison. (a) We study the influence of data augmentation on model performance. (b) Conduct data augmentation on different models to evaluate the performance of model.

4.6 Ablation Studies

In this section, we conduct ablation studies to evaluate the effect of each component in DAKT.

Main Ablation Studies. The extra performance gains from DAKT can be summarized into two aspects, data augmentation and the network structure. In particular, the two aspects may intersect, that is, a well-designed network structure could take full advantage of using the information of presentations from the generative data. We first study the extent of the generated data influences on the model performance. We generate data that half of the original data and input it into the network together with the original data. For comparison, we do not use any generated data, and directly use original data as input for the model. As shown in Fig. 4(a), using generated data improves performance of KT in each dataset, but the improvement varies. Then, we study the influence of different networks on KT performance to prove the effectiveness of our model. With generated data which accounts for 50% of the original data being fed into the training set, we take observation of the performance by the DKT model (classical method) and the ATKT model (relatively well performed method). As shown in Fig. 4(b), our model outperforms the other two with an increase of AUC scale on every dataset. It is easy to conclude that the structure of KT network is also crucial for the task. More importantly, data augmentation and sophisticated network are complementary to improve the KT performance.

Ablation Studies on Data Augmentation. The effectiveness of data augmentation method is affected by the proportion of generated data in input data and the training rounds of VAE model. To supplement, because of the introduction of generated data increases the risk on drift of data fitting, regularization measures (in this paper, dropout) should be taken. We conduct ablation experiments of data augmentation from three comparative aspects (without generation vs. 5% generation rate, number of VAE training epochs 3 vs. 15, with dropout strategy vs. without dropout strategy). As mentioned above, the proportion of generated data has little impact on KT performance, so we set the generation rate at 5% for convenience. The VAE training epochs means the degree of discrimination between the generated data and the original data, and it is fully trained by 15 epochs while 3 epochs not. As shown in Table 5, it can be found that DAKT performs 1.34% higher than the best previous work in the scale of average AUC on the four datasets even without data augmentation. With data augmentation, dropout strategy is more important than the VAE training epochs, which denotes that even if the generative model is not fully trained, it can still produce effective representations. Meanwhile, the regularization trick of dropout can improve the performance of the model by reducing the fitting drift caused by the generated data.

Table 5. Ablation studies on data augmentation.

Generation rate		VAE training epochs		Dropout strategy		AUC on average
0	5%	3	15	on	off	
✓						81.12%
	✓	✓			✓	81.51%
	✓	✓		✓		82.01%
	✓		✓		✓	81.78%
	✓		✓	✓		82.19%

Ablation Studies on KT Network. In order to research which module in the network is more important for the performance of KT, we combine different modules without data augmentation and test the average AUC on the four datasets. As shown in Table 6, each component contributes to KT’s performance, but the contribution varies. The tricks of optimize KT network can be summarized as follows. Firstly, long-term connection mechanism of attention module could helpful to improve the performance of the model because it extracts the important presentation information in each layer and propagates it forward. Secondly, LSTM module is an essential module, because as a recurrent neural network (RNN), it can capture the time series information of the input sequences. Last but not least, the concatenation trick avoids the risk of other modules discarding information, and it could keep the original information when used with other modules.

Table 6. Ablation studies on KT network.

Attention module 1 + 2	LSTM module	Concatenation trick	Attention module 3	AUC on average
✓				73.96%
✓			✓	74.80%
	✓			77.43%
✓		✓	✓	75.64%
	✓	✓		79.38%
✓	✓		✓	80.92%
✓	✓	✓	✓	81.12%

5 Conclusion

In this paper, we have proposed a novel approach named DAKT to overcome the data scarcity in KT task. We generate question sequences from VAE and answer label sequences from pre-trained KT network respectively, and combine them to get generated data. With generated data in training set, we have verified through experiments that DAKT is less dependent on the amount of data, and it can improve the performance of the model in KT task. Compared with several state-of-the-art models, our method outperforms the others on four typical datasets and achieves an improvement of 2.41% more than the suboptimal one on average AUC performance. To evaluate the effect of each component in DAKT, we perform in-depth ablation studies and in summary, data augmentation strategy and appropriate design of the model proved to be key factors for improving KT performance.

Acknowledgments. This work was supported by the National Natural Foundation of China under Contract 62071014.

References

1. Liu, Q., Shen, S., Huang, Z., Chen, E., Zheng, Y.: A survey of knowledge tracing. arXiv preprint [arXiv:2105.15106](https://arxiv.org/abs/2105.15106) (2021)
2. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems (2015)
3. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)
4. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint [arXiv: 1907.06837](https://arxiv.org/abs/1907.06837) (2019)
5. Liu, Q., et al.: EKT: exercise-aware knowledge tracing for student performance prediction. IEEE Trans. Knowl. Data Eng. **33**(1), 100–115 (2019)

6. Su, Y., et al.: Exercise-enhanced sequential modeling for student performance prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
7. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2330–2339 (2020)
8. Huang, Z., et al.: Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students. ACM Trans. Inf. Syst. (TOIS) **38**(2), 1–33 (2020)
9. Khosravi, H.: Recommendation in personalised peer-learning environments. arXiv preprint [arXiv:1712.03077](https://arxiv.org/abs/1712.03077) (2017)
10. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian knowledge tracing models. In: Chad Lane, H., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18
11. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the Fifth Annual ACM Conference on Learning at Scale, pp. 1–10 (2018)
12. Guo, X., Huang, Z., Gao, J., Shang, M., Shu, M., Sun, J.: Enhancing knowledge tracing via adversarial training. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 367–375 (2021)
13. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 175–184 (2019)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
15. Khajah, M., Lindsey, R.V., Mozer, M.C.: How deep is knowledge tracing?. arXiv preprint [arXiv:1604.02416](https://arxiv.org/abs/1604.02416) (2016)
16. Pandey, S., Srivastava, J.: RKT: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1205–1214 (2020)
17. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-Adap. Inter. **4**, 253–278 (1994)
18. Pardos, Z., Bergner, Y., Seaton, D., Pritchard, D.: Adapting Bayesian knowledge tracing to a massive open online course in edX. In: Educational Data Mining 2013 (2013)
19. Liu, Q., et al.: Exploiting cognitive structure for adaptive learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 627–635 (2019)
20. Schodde, T., Bergmann, K., Kopp, S.: Adaptive robot language tutoring based on Bayesian knowledge tracing and predictive decision-making. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pp. 128–136 (2017)
21. Surjono, H.D.: The evaluation of a moodle based adaptive e-learning system. Int. J. Inf. Educ. Technol. **4**(1), 89 (2014)
22. Desmarais, M.C., Baker, R.S.D.: A review of recent advances in learner and skill modeling in intelligent learning environments. User Model. User-Adap. Inter. **22**, 9–38 (2012)
23. Kasurinen, J., Nikula, U.: Estimating programming knowledge with Bayesian knowledge tracing. ACM SIGCSE Bull. **41**(3), 313–317 (2009)
24. Huang, Z., et al.: Exploring multi-objective exercise recommendations in online education systems. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1261–1270 (2019)
25. Chabbott, C., Ramirez, F.O.: Development and education. In: Hallinan, M.T. (ed.) Handbook of the Sociology of Education, pp. 163–187. Springer, Cham (2006). https://doi.org/10.1007/0-387-36424-2_8

26. Wotto, M.: The future high education distance learning in Canada, the United States, and France: insights from before COVID-19 secondary data analysis. *J. Educ. Technol. Syst.* **49**(2), 262–281 (2020)
27. Suo, Y., Miyata, N., Morikawa, H., Ishida, T., Shi, Y.: Open smart classroom: extensible and scalable learning system in smart space using web service technology. *IEEE Trans. Knowl. Data Eng.* **21**(6), 814–828 (2008)
28. Nguyen, T.: The effectiveness of online learning: beyond no significant difference and future horizons. *MERLOT J. Online Learn. Teach.* **11**(2), 309–319 (2015)
29. Emanuel, E.J.: MOOCs taken by educated few. *Nature* **503**(7476), 342 (2013)
30. VanLehn, K.: The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **46**(4), 197–221 (2011)
31. Cully, A., Demiris, Y.: Online knowledge level tracking with data-driven student models and collaborative filtering. *IEEE Trans. Knowl. Data Eng.* **32**(10), 2000–2013 (2019)
32. Li, C., Zhou, P., Xiong, L., Wang, Q., Wang, T.: Differentially private distributed online learning. *IEEE Trans. Knowl. Data Eng.* **30**(8), 1440–1453 (2018)
33. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
34. Doersch, C.: Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016)
35. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International Conference on Machine Learning, pp. 1530–1538 (2015)
36. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017)
37. Pu, Y., et al.: Variational autoencoder for deep learning of images, labels and captions. In: Advances in Neural Information Processing Systems (2016)
38. Caterini, A.L., Doucet, A., Sejdinovic, D.: Hamiltonian variational auto-encoder. In: Advances in Neural Information Processing Systems (2018)
39. Shen, S., et al.: A survey of knowledge tracing: models, variants, and applications. *IEEE Trans. Learn. Technol.* (2024)
40. Kuo, M., Sarker, S., Qian, L., Fu, Y., Li, X., Dong, X.: Enhancing deep knowledge tracing via diffusion models for personalized adaptive learning. *arXiv preprint arXiv:2405.05134* (2024)



An Epidemic Trend Prediction Model with Multi-source Auxiliary Data

Benfeng Wang, Xiaohua He, Hang Lin, Guojiang Shen, and Xiangjie Kong^(✉)

Zhejiang University of Technology, Hangzhou 310014, China
xjkong@ieee.org

Abstract. The global outbreak of epidemics profoundly affects public health and societal development. The development of epidemic trend prediction models is crucial to prevent the recurrence of pandemics. Therefore, we propose a Bayes-Attention AL-Forecast prediction (BALF) model to forecast the future development trends of epidemics. Firstly, we introduce an attention mechanism to integrate the population mobility data features into the case data. Subsequently, based on fused data, we employ an ARIMA-LSTM Forecast (AL-Forecast) model to predict the development trends of epidemics. Finally, experiments are conducted based on real datasets. The results indicate a close correlation between predicted and actual case numbers, and the model's prediction performance excels with baseline and other state-of-the-art methods. We release our source code at <https://github.com/Bevan-Wang/MEHP>.

Keywords: Epidemic trend forecasting · Bayes-Attention mechanism · Hybrid forecasting model · Multi-source data fusion

1 Introduction

The World Health Organization (WHO) introduced the concept of Disease X in 2018 and has since issued multiple warnings urging readiness for the next occurrence of Disease X. Disease X isn't a specific ailment but rather an infectious disease caused by an unknown pathogen capable of triggering a global pandemic. It holds the potential to emerge unpredictably from various sources, posing a grave threat to millions of lives and inflicting substantial losses on communities, nations, businesses, and economies. This presents a formidable challenge to epidemic prevention and control efforts, underscoring the necessity of investing in developing preventive models for future pandemics [5].

Accurately predicting the future trends of epidemics has become a significant focus of global scholars. Most work mainly involves two approaches: traditional mathematical models and deep learning models. For example, Kong [14] employed the ARIMA model to interpolate the incidence data of dengue fever in Zhejiang Province over the past 15 years and established the relationship between dengue fever incidence and meteorological factors using the LSTM

model. Xue et al. [26] integrated GNN and LSTM networks to model the intricate relationships among urban areas, inter-regional mobility patterns, network search history, and future COVID-19 infections. Their model accurately predicted Multiwave COVID-19 trends. Konwer et al. [15] proposed an attention-based multi-scale gated recurrent encoder to predict the progression of lung infiltrates. Cai et al. [4] introduced an end-to-end network based on a Transformer for automatically detecting COVID-19. Traditional mathematical models are proficient at handling linear tasks in data, such as long-term dependencies and periodicity. In contrast, deep learning models are crucial in capturing more complex nonlinear patterns. The spread of epidemics entails intricate scenarios, presenting challenges of long-term dependencies and dynamic complex changes. Therefore, integrating traditional mathematical models and deep learning models holds significant research significance for predicting epidemic development trends.

The data sources for epidemic forecasting tasks are diverse, encompassing case data, sentiment text data, and travel mobility data. For example, Wang et al. [24] introduced a novel deep learning approach driven by epidemic case data, employing black-box knowledge distillation to achieve accurate and efficient transfer dynamic predictions practically. Min et al. [19] proposed a machine reading system (ExcavatorCovid) utilizing text corpora to assist governments in managing information overload and mitigating the impact of the pandemic. Hao et al. [9] utilized real-world mobility data to understand the urban epidemic spread of COVID-19. Shahid et al. [22] utilized models such as LSTM, GRU, and Bi-LSTM to forecast epidemic trends, encompassing confirmed, death, and recovery cases. Case data often only provides short-term information, posing challenges to prediction accuracy. Auxiliary data sources such as sentiment text and population mobility can indirectly reflect epidemic transmission characteristics but require further exploration of their relevance. Fusing auxiliary data sources as supplements of case data to enhance epidemic prediction performance has become a new challenge.

Therefore, we aim to fuse multi-source epidemic data while predicting future epidemic trends. To address this issue, we propose a **Bayes-Attention AL-Forecast Prediction (BALF)** model. First, we introduce a Bayes-Attention mechanism to integrate multi-source epidemic data. Secondly, we propose an ARIMA-LSTM Forecast (AL-Forecast) model to predict the future development trend of the epidemic cases. Finally, we conduct comparison and ablation experiments based on real datasets. Our main contributions are threefold:

- We propose an AL-Forecast model to predict future epidemic trends, addressing the long-term dependency and dynamic complexity inherent in epidemic transmission.
- We design a multi-source data fusion method based on a Bayes-Attention mechanism to enhance the prediction accuracy.
- We conduct experiments on a real epidemic dataset. The results show that predicted confirmed cases are in good agreement with actual confirmed cases,

and the prediction performance is higher than baseline methods and other state-of-the-art methods.

2 Related Work

2.1 Epidemic Prediction Models

Researchers have applied various techniques to forecast epidemic patterns, including traditional mathematical models [16] and deep learning models [6]. Conventional mathematical models excel at handling linear and periodic tasks. For example, Zhang et al. [27] proposed a dual-granularity directional representation approach for predicting infectious disease cases, effectively learning temporal dynamics from dual-granularity time series data. To predict the spread of COVID-19 from large-scale mobility data, Schwabe et al. [21] introduced a novel epidemiological forecasting model based on mobility data called the Mobility-Labeled Hawkes model. This model utilizes Hawkes processes to capture disease transmission dynamics and makes predictions based on regularized Poisson regression. Abdullahzazak et al. [3] employed the ARIMA model to forecast confirmed and recovered cases at various stages of Kuwait’s phased prevention plan. While these studies forecast future epidemic trends, the accuracy of epidemic prediction could be higher and may not be suitable for complex and dynamic environments.

Deep learning models have demonstrated remarkable capabilities in pattern recognition and prediction. For instance, Cui et al. [7] proposed an encoder-decoder framework for predicting the number of epidemic cases and deaths. Kim et al. [12] developed a fine-grained economic epidemiological modeling framework named COVID-EENet, featuring a two-tiered deep neural network, to predict the localized economic impact of COVID-19. To forecast multi-wave pandemics, Xue et al. [26] introduced a Social-aware Graph Neural Network (SAB-GNN) that considers the decay of symptom-related network search frequencies to capture changes in public awareness during multi-wave pandemics. To forecast the number of inbound COVID-19 cases in South Korea, Kim et al. [13] proposed a neural network named Hi-COVIDNet, which incorporates a two-tiered encoder architecture. To predict the dynamically unstable COVID-19 trends across multiple regions, Zhang et al. [28] introduced a novel spatiotemporal forecasting framework called HIERST. This architecture integrates various deep learning techniques such as CNNs, RNNs, and GCNs to enhance the robustness of the model predictions. While these studies have achieved predictions of specific epidemic case numbers, they are primarily suited for handling short-term volatile tasks. They may need to provide advantages for long-term dependent tasks.

As a result, researchers have explored the application of conventional mathematical and deep-learning hybrid models in epidemic prediction. Jin et al. [11] introduced a parallelized approach based on a regression coefficient-weighted ARIMA-LSTM model to forecast epidemic data in China. Li et al. [17] proposed a novel hybrid forecasting model named GVMD-ELM-ARIMA, based on Gradient Variational Mode Decomposition (GVMD), Extreme Learning Machine

(ELM), and Autoregressive Integrated Moving Average (ARIMA), to enhance the prediction accuracy of cumulative COVID-19 confirmed cases. These hybrid methods are more accurate than independent conventional mathematical models and deep learning models. Therefore, efforts are underway to enhance prediction accuracy by integrating traditional mathematical models and deep learning models.

2.2 Multi-source Fusion

Most epidemic prediction tasks rely on case data. For example, Rodriguez et al. [20] proposed a neural transfer learning architecture named CALI-Net based on COVID-19 and influenza cases, applying historical forecasting models to the current pandemic situation. Although case data can directly reflect the transmission characteristics of the epidemic, the sparsity of data limits the accuracy of predictions. Therefore, multi-source auxiliary data is beginning to be applied in epidemic tasks. For example, Xiao et al. [25] leverage extensive long-term human mobility data collected from Baidu Maps to proactively screen each community in target cities and predict infection risks. Drinkall et al. [8] utilized unsupervised embeddings of social media posts to predict clusters of COVID-19 case counts. While these data sources may not directly reflect epidemic developments, they can serve as supplementary tools for evaluation and prediction.

Therefore, researchers have shifted their focus to integrating multi-source epidemic data. For example, Adiga et al. [2] employed the transfer entropy technique and phase inference method to integrate auxiliary data sources and developed the Bayesian model averaging ensemble technique [1], further enhancing the fusion effect. Trajanoska et al. [23] predicted the mortality rate of COVID-19 by integrating dietary, comorbidity, and geo-economic data. Lv et al. [18] employed a matrix data fusion module to perform a multiscale fusion of COVID-19 epidemic data and traffic revitalization index data, further enhancing the model's prediction performance. Inspired by their work, we will integrate multi-source data, incorporating auxiliary data sources into epidemic case data to improve prediction accuracy.

3 Methodology

This section will provide a detailed description of our proposed method. Our approach involves developing a Bayes-Attention AL-Forecast Prediction (BALF) model to integrate multi-source epidemic data to forecast case trends. Firstly, we introduce a Bayes-Attention mechanism to fuse multi-source data. This mechanism integrates various indirect data sources into the case data based on dynamic weights. Secondly, we employ the AL-Forecast model to predict case trends. Combining the strengths of ARIMA and LSTM models, the AL-Forecast model effectively handles both linear and nonlinear tasks, addressing challenges related to long-term dependency and dynamic complexity in epidemic transmission. Finally, we assess the prediction performance using a range of evaluation metrics.

Specifically, we will describe the proposed method from three perspectives: the Bayesian Attention Mechanism, the AL-Forecast Model, and evaluation metrics. The framework structure of the model is illustrated in Fig. 1.

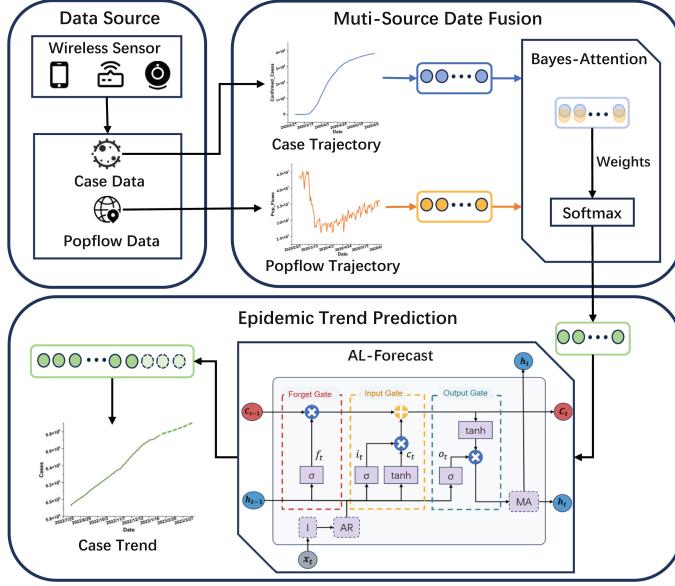


Fig. 1. The framework structure of the BALF model.

3.1 Bayes-Attention

The Bayes-Attention mechanism is a data fusion technique of information-weighted aggregation, dynamically assigning weights to various information sources based on their contextual relevance and significance. This mechanism plays a crucial role in efficiently integrating multi-source data. First, the attention mechanism is used to learn the dynamic allocations of weight between multi-source data adaptively. Subsequently, leveraging Bayesian networks, fused data sequences are generated by incorporating weighted fusion conditional probability distributions.

Let $X = \{X_1, X_2, \dots, X_T\}$ be the connection vector generated from multi-source data. The data fusion process based on the Bayes-Attention mechanism can be expressed as follows:

$$M = \tanh(X), \quad (1)$$

$$\alpha = \text{softmax}(w^T \cdot M), \quad (2)$$

$$P_i = \frac{\alpha_i \cdot P(X_i | X)}{\sum_{i=1}^T \alpha_i \cdot P(X_i | X)}, \quad (3)$$

$$p = \sum_{i=1}^T X_i \cdot P^T. \quad (4)$$

In the above equation, w represents the learned parameter vector, which maps the hidden states to attention weights. α denotes the attention weights, and P signifies the conditional probability distribution.

Finally, to facilitate comparisons with other baseline and state-of-the-art models in the experimental section, We process the fused dataset through a *softmax* function to concentrate its values within the range $[0, 1]$.

$$x = \text{softmax}(p). \quad (5)$$

3.2 AL-Forecast

The ARIMA-LSTM Forecast (AL-Forecast) model is used to predict the epidemic development trend. To address the long-term dependency and dynamic complexity issues in epidemic spread, we consider integrating traditional mathematical models and deep learning models to enhance prediction accuracy. Firstly, we stabilize the sequence through differencing operations. Secondly, we use autoregressive and LSTM models to predict future values based on historical data. Finally, we employ moving average models to predict based on past error terms, improving prediction accuracy.

Differential operation can transform a non-stationary sequence into a stationary one, typically involving first-order or higher-order differences. By visualizing the differentials of the sequence, we can determine its order d .

Since both autoregressive models and LSTM models predict future values based on past historical data, we consider integrating them organically. The autoregressive model relies on the relationship between historical and current values, thus demonstrating significant advantages in long-term dependency prediction tasks. To evaluate the effect of order q , we introduce the Partial Auto-Correlation Function (PACF). When the value approaches 1, it indicates a high positive correlation between the observations at two-time points. When the value approaches -1 , it indicates a high negative correlation between the observations at two-time points. When the value approaches 0, it means a weak correlation between the observations at two time points. The calculation process of the autoregressive model is as follows:

$$y_t = \mu + \sum_{i=1}^p \gamma_i y_{t-i} + \varepsilon_t, \quad (6)$$

$$E(t) = E[X_t | X_{t-1}, \dots, X_{t-d+1}], \quad (7)$$

$$PACF = \frac{\text{Cov}(X_t - E(t), X_{t-d} - E(t-d))}{\text{Var}(X_t)}. \quad (8)$$

In the above equation, y_t is the current value, μ is a constant term, p is the order, θ_i represents the autoregressive coefficients, and ε_t is the error term. Cov represents the covariance of the observed values, E denotes the expectation, which is the mean value, and Var represents the time series variance.

LSTM model integrates a state structure and three gate structures: the cell state, forget gate, input gate, and output gate, facilitating the dynamic adjustment of self-recurrent weights. The forget gate determines which information from the previous time step's cell state should be overlooked or discarded. It utilizes a sigmoid activation function to produce an output between 0 and 1, where 0 signifies complete forgetting, and 1 indicates complete retention. The input gate regulates the extent to which new input data flows in. It incorporates a sigmoid activation function and its output ranges between 0 and 1. If the output of the input gate is close to 1, the network considers this information necessary and should be incorporated into the cell state. The output gate regulates the output from the cell to the hidden state. It employs a sigmoid activation function to determine which parts of the cell state should be output. Simultaneously, it uses a tanh activation function to ensure that the output values h_t are within the range of -1 to 1. The calculation process for the LSTM model is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (9)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (10)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (11)$$

$$C_t = f_t * C_{t-1} + i_t * c_t, \quad (12)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (13)$$

$$h_t = o_t * \tanh(C_t). \quad (14)$$

In the above equation, x_t is the input, h_{t-1} is the previous time step's hidden state, and h_t is the final output. f_t , i_t , and o_t is the output of each gate, σ is the sigmoid activation function, and c_t is the candidate cell state. W is the weight matrix and b is the bias term.

The moving average model can further improve prediction effectiveness. It represents the relationship between the current value and the linear of past errors, focusing on the accumulation of residual terms. The order q compares the current value with the linear combination of past q error terms. To evaluate the effectiveness of the order q , we introduce the Auto-Correlation Function (ACF). Its evaluation criteria are consistent with the PACF. The calculation process of the moving average model is as follows:

$$y_t = \mu + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t, \quad (15)$$

$$ACF = \frac{Cov(X_t, X_{t-d})}{Var(X_t)}. \quad (16)$$

In the above equation, y_t is the current value, μ is a constant term, q is the order, θ_i represents the moving average coefficients, and ε is the error term.

We compute the Bayesian Information Criterion (BIC) to determine the optimal model. The BIC calculates the probability function and adds a penalty term

for the number of parameters, which helps to avoid overfitting and provides a balanced approach to model selection. The calculation formula is described as

$$BIC = k \ln(n) - 2 \ln(L), \quad (17)$$

where k is the number of parameters, n is the sample size, L is the likelihood function. A smaller BIC value indicates that the given parameters provide a more accurate model description. The differencing order d can be determined through visualization methods by selecting the order that minimizes the fluctuations in the time series.

3.3 Evaluation Metrics

To compare the prediction effects of different models more intuitively, we use the root mean squared error (RMSE), the mean absolute percentage error (MAPE), the mean absolute error (MAE), and the r-squared score(R^2).

RMSE: Root Mean Squared Error (RMSE) is the square root of the mean of the squared differences between predicted and actual values. It demonstrates a remarkable sensitivity to prediction errors, exerting a more substantial penalty on significant errors than minor discrepancies. The formula for calculating RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}. \quad (18)$$

MAPE: Mean Absolute Percentage Error (MAPE) represents the average percentage error of each observed value. It expresses errors in percentage form, making it more easily understandable. The calculation formula for MAPE is as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right|. \quad (19)$$

MAE: Mean Absolute Error (MAE) is the average of the absolute differences between predicted and actual values. Unlike RMSE, MAE does not consider the square of errors, making its treatment of large and small errors relatively equal. The formula for calculating MAE is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (20)$$

R² : R-squared (R^2) represents the extent to which the model explains the variability of the target variable. It ranges from 0 to 1, with values closer to 1 indicating a better fit of the model. An R^2 of 1 signifies a perfect fit to the data, while an R^2 of 0 implies that the model cannot explain the variability of the target variable. The calculation formulas for R^2 is as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}. \quad (21)$$

In the above formulas, n is the total sample size, \hat{y}_i is the predicted value of the model, y_i is the true value, \bar{y} is the average of the true value. Lower values indicate better performance for RMSE, MAPE, and MAE, approaching 0. On the other hand, for R^2 , a value closer to 1 signifies better model fit and explanatory power.

4 Experiment

4.1 Datasets

Our proposed model experiments were completed in Python 3.11.6. This study utilizes the cumulative confirmed case numbers in New York State from March 2, 2020, to February 24, 2023, and daily population mobility trajectory numbers from February 15, 2020, to October 15, 2022, for modeling, prediction, and analysis. The data were sourced from the Johns Hopkins University website and the U.S. COVID-19 pandemic population mobility dataset, both publicly available. The data sets are shown in Table 1 and Table 2, respectively.

Table 1. The cumulative confirmed cases dataset

Province/State/Time	4/25/21	4/26/21	4/27/21	4/28/21	4/29/21	4/30/21
Alabama/Cases	526131	526348	526707	527083	527513	527922
California/Cases	3794549	3796285	3798103	3801766	3799797	3804036
New York/Cases	2031093	2034102	2037414	2044345	2040448	2048150
Washington/Cases	395312	397417	398509	401718	400149	403040

Table 2. The daily population mobility dataset

$geoid_o$	$geoid_d$	lng_o	lat_o	lng_d	lat_d	$date_range$	pop_flows
1001	1001	-86.64	32.53	-86.64	32.53	03/09/20–03/15/20	86486
1001	1015	-86.64	32.53	-85.83	33.77	03/09/20–03/15/20	562
1001	1051	-86.64	32.53	-86.15	32.60	03/09/20–03/15/20	147182
1001	1007	-86.64	32.53	-87.13	33.00	03/09/20–03/15/20	84

Upon acquiring the data, the outliers were cleaned through time series diagrams, and mean substitution treatment was applied for outlier handling. We utilize the first 80% of the data as the training set for the model and the remaining 20% as the test set to assess the model's generalization ability.

4.2 BALF Model

Here, experimental parameters were determined, and prediction results for future epidemic trends were obtained.

Initially, the original data undergoes differencing. After applying a second-order difference, the signal sequence becomes stationary and white noise-free. To determine the parameters of ARIMA(p, d, q), we calculate the Bayesian Information Criterion (BIC) for evaluation. The results of the BIC are illustrated in a heatmap, as shown in Fig. 2 (a).

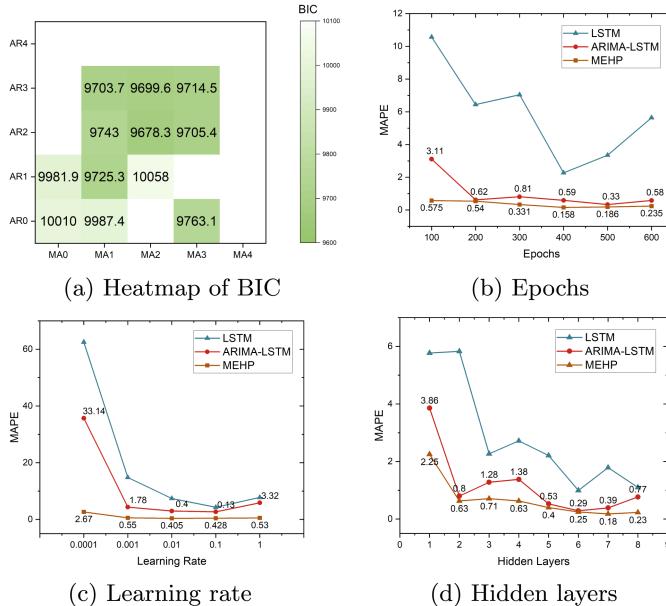


Fig. 2. Hyperparameter experiments.

From the heat map, it can be determined that the order of the autoregressive model is 2, and the order of the moving average is 2. To determine the parameters of the LSTM model, we conduct hyperparameter experiments. The process of training model hyperparameters is illustrated in Fig. 2 (b)–(d).

Due to MAPE representing the mean percentage error and exhibiting superior performance, we utilize MAPE to assess the training effectiveness of the

Table 3. The parameters of the BALF model

Parameter	Value
(p, d, q)	(2, 2, 2)
Hidden layers	7
layer_num	2
Epochs	400
Learning rate	0.01

model. The parameters for the BALF model are configured as specified in Table 3. Figure 3 (a) shows the time series graph comparing the predicted values.

4.3 Comparison Experiment

We introduce several baseline and state-of-the-art models to provide a more precise comparison of the prediction performance. These time series forecasting models include LSTM, Bi-LSTM, GRU, Transformer, and CNN-ARIMA-LSTM.

LSTM: Shahid’s team utilized LSTM [22], GRU, and Bi-LSTM models to forecast the time series of confirmed cases, deaths, and recoveries. We implement the model methodologies from their work and perform prediction tasks based on our dataset.

Bi-LSTM: Bidirectional Long Short-Term Memory (Bi-LSTM) [22] is an extension of the Long Short-Term Memory (LSTM) network designed for processing sequential data. It introduces two independent LSTM layers at each time step: one for forward propagation and the other for backward propagation.

GRU: Gated Recurrent Unit (GRU) [22] is a variant of Recurrent Neural Network (RNN) designed for processing sequential data. It controls the flow and update of information in sequence data through gate mechanisms, including update and reset gates, thereby modeling long-term dependencies. The GRU model has fewer parameters, faster training speed, and better generalization capabilities.

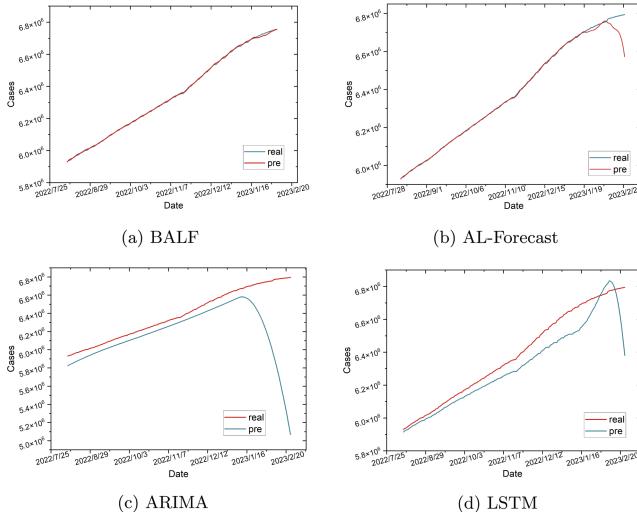


Fig. 3. Time series diagram comparing the predicted and true values of the BALF, AL-Forecast, ARIMA, and LSTM model.

Transformer: Transformer is a deep learning model based on attention mechanisms, with its core ideas including the self-attention mechanism and positional encoding. Its modular structure and parallel computation capabilities address the efficiency issues of recurrent neural networks (RNNs) when dealing with long sequences.

CNN-ARIMA-LSTM: Jin et al. [10] proposed three combined forecasting models, namely CNN-LSTM-ARIMA, TCN-LSTM-ARIMA, and SSA-LSTM-ARIMA, to predict the pandemic in Italy. Ultimately, they demonstrated that the CNN-LSTM-ARIMA model performed the best. Therefore, we conduct comparative experiments using the CNN-LSTM-ARIMA model.

We evaluate the performance of each model using evaluation metrics such as the root mean squared error (RMSE), the mean absolute percentage error (MAPE), the mean absolute error (MAE), and the determination coefficient (R^2). The prediction performance evaluation metrics of each model on the test set are shown in Table 4.

Table 4. Comparison experiment results

Model	RMSE	MAPE	MAE	R^2
LSTM	0.078	8.49	0.076	0.089
Bi-LSTM	0.058	5.92	0.053	0.524
GRU	0.063	0.56	0.057	0.957
Transformer	0.145	15.21	0.143	0.544
CNN-ARIMA-LSTM	0.126	6.75	0.124	0.285
BALF (Ours)	0.025	0.62	0.002	0.997

Based on the four evaluation metrics selected for this study, the performance of the relevant models was compared. We use bold to denote the best performance indicators. Among them, the BALF model shows the best prediction performance. Note that the GRU model achieves the best performance in the MAPE metric. However, the BALF model still exhibits good performance.

4.4 Ablation Study

We conduct ablation experiments to validate each module's contribution to our proposed model. Firstly, for the case data without data fusion, we utilize the AL-Forecast model for prediction tasks to verify the effectiveness of the Bayes-Attention mechanism. Subsequently, prediction tasks are conducted separately using independent ARIMA and LSTM models to validate the enhancement of prediction accuracy by the AL-Forecast model. The ARIMA model here refers to removing the LSTM module from the original AL-Forecast model. Finally, we compare the prediction performance metrics among the four components.

AL-Forecast Model: We retrain the model and visualize the training process of hyperparameters in Fig. 2. The parameters for the hybrid model are configured as specified in Table 5.

Table 5. The parameters of the AL-Forecast model

Parameter	Value
(p, d, q)	(2, 2, 2)
Hidden layers	6
layer_num	2
Epochs	500
Learning rate	0.1

Once the parameters are configured, we train the AL-Forecast model using the training set and then utilize the trained model to make predictions on the test set. The time series graph comparing the predicted values of the AL-Forecast model with the actual values is illustrated in Fig. 3 (b).

ARIMA Model: Using the Bayesian Information Criterion (BIC), the parameters of the model (p, d, q) are determined to be (2, 2, 2). The comparison between the ARIMA model's predicted values and the actual values in the time series is depicted in Fig. 3 (c).

LSTM Model: To determine the parameters of the LSTM model, we conducted hyperparameter experiments, as shown in Fig. 2. The model parameters are configured as specified in Table 6. The time series graph comparing the predicted values of the LSTM model with the actual values is shown in Fig. 3 (d).

Table 6. The parameters of the LSTM model

Parameter	Value
Hidden layers	6
layer_num	2
Epochs	400
Learning rate	0.1

According to the comparative analysis of prediction results from various models, it is evident that the BALF model indeed exhibits superior forecasting accuracy. We still use evaluation metrics such as RMSE, MAPE, MAE, and R^2 to

Table 7. Ablation experiment results

Model	RMSE	MAPE	MAE	R ²
ARIMA	0.066	4.83	0.044	0.396
LSTM	0.078	8.49	0.076	0.089
AL-Forecast	0.029	0.93	0.008	0.882
BALF (Ours)	0.025	0.62	0.002	0.997

compare more precisely. The performance evaluation metrics for each model on the test set are presented in Table 7.

Based on four evaluation criteria, we compared the performance of various modules in our model. Among them, the BALF model exhibited the best prediction performance, and the prediction performance of the AL-Forecast model surpassed that of independent LSTM or ARIMA models. Experimental results demonstrate that our proposed Bayes-Attention mechanism and the AL-Forecast prediction model significantly enhance prediction accuracy.

5 Conclusion

Epidemic data show high dynamics and seasonality, and the data sources are complex, which poses a significant challenge for prediction work. To predict the case trend of the epidemic more accurately, we propose a Bayes-Attention AL-Forecast prediction (BALF) model. Initially, we introduce a Bayes-Attention mechanism to fuse multi-source data features dynamically. Subsequently, we propose a hybrid time series prediction model, namely AL-Forecast, to predict the future case trend of the epidemic. In general, the calculated metrics of the BALF model are as follows: RMSE = 0.025, MAPE = 0.62, MAE = 0.002, R² = 0.997. The prediction performance of the BALF model exceeds that of the AL-Forecast model, and the prediction performance of the AL-Forecast model exceeds that of the stand-alone LSTM or ARIMA model. Therefore, the Bayes-Attention fusion mechanism and the AL-Forecast hybrid model can help predict the epidemic's future trend more accurately.

References

1. Adiga, A., et al.: Enhancing covid-19 ensemble forecasting model performance using auxiliary data sources. In: 2022 IEEE International Conference on Big Data (Big Data), pp. 1594–1603. IEEE (2022)
2. Adiga, A., et al.: All models are useful: Bayesian ensembling for robust high resolution covid-19 forecasting. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2505–2513 (2021)
3. Alabdulrazzaq, H., Alenezi, M.N., Rawajfih, Y., Alghannam, B.A., Al-Hassan, A.A., Al-Anzi, F.S.: On the accuracy of Arima based prediction of Covid-19 spread. Results Phys. **27**, 104509 (2021)

4. Cai, C., Liu, B., Tao, J., Tian, Z., Lu, J., Wang, K.: End-to-end network based on transformer for automatic detection of Covid-19. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9082–9086. IEEE (2022)
5. Caulkins, J., et al.: The hammer and the jab: are covid-19 lockdowns and vaccinations complements or substitutes? *Europ. J. Oper. Res.* (2023)
6. Chimmula, V.K.R., Zhang, L.: Time series forecasting of Covid-19 transmission in Canada using LSTM networks. *Chaos, Solitons Fractals* **135**, 109864 (2020)
7. Cui, Y., Zhu, C., Ye, G., Wang, Z., Zheng, K.: Into the unobservables: a multi-range encoder-decoder framework for Covid-19 prediction. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 292–301 (2021)
8. Drinkall, F., Zohren, S., Pierrehumbert, J.: Forecasting COVID-19 caseloads using unsupervised embedding clusters of social media posts. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1471–1484 (2022)
9. Hao, Q., Chen, L., Xu, F., Li, Y.: Understanding the urban pandemic spreading of covid-19 with real world mobility data. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 3485–3492 (2020)
10. Jin, Y.C., et al.: Models for Covid-19 data prediction based on improved LSTM-arima algorithms. *IEEE Access* (2023)
11. Jin, Y., et al.: Prediction of Covid-19 data using an Arima-LSTM hybrid forecast model. *Mathematics* **10**(21), 4001 (2022)
12. Kim, D., et al.: Covid-eenet: Predicting fine-grained impact of Covid-19 on local economies. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 11971–11981 (2022)
13. Kim, M., et al.: Hi-Covidnet: Deep learning approach to predict inbound Covid-19 patients and case study in south korea. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 3466–3473 (2020)
14. Kong, L.: Autoregressive moving average model and improved LSTM neural network applied in epidemic prediction in Zhejiang province. *J. Phys.: Conf. Series*. vol. 2033, p. 012104. IOP Publishing (2021)
15. Knower, A., et al.: Attention-based multi-scale gated recurrent encoder with novel correlation loss for COVID-19 progression prediction. In: de Bruijne, M., et al. (eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27 – October 1, 2021, Proceedings, Part V*, pp. 824–833. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-87240-3_79
16. Kumar, S., Viral, R.: Effect, challenges, and forecasting of Covid-19 situation in India using an Arma model. *IEEE Trans. Comput. Social Syst.* **8**(4), 955–963 (2021)
17. Li, G., Chen, K., Yang, H.: A new hybrid prediction model of cumulative Covid-19 confirmed data. *Process Saf. Environ. Prot.* **157**, 1–19 (2022)
18. Lv, Z., Wang, X., Cheng, Z., Li, J., Li, H., Xu, Z.: A new approach to Covid-19 data mining: a deep spatial-temporal prediction model based on tree structure for traffic revitalization index. *Data Knowl. Eng.* **146**, 102193 (2023)
19. Min, B., Rozonoyer, B., Qiu, H., Zamanian, A., MacBride, J.: Excavatorcovid: Extracting events and relations from text corpora for temporal and causal analysis for Covid-19. arXiv preprint [arXiv:2105.01819](https://arxiv.org/abs/2105.01819) (2021)

20. Rodríguez, A., Muralidhar, N., Adhikari, B., Tabassum, A., Ramakrishnan, N., Prakash, B.A.: Steering a historical disease forecasting model under a pandemic: Case of flu and Covid-19. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4855–4863 (2021)
21. Schwabe, A., Persson, J., Feuerriegel, S.: Predicting Covid-19 spread from large-scale mobility data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3531–3539 (2021)
22. Shahid, F., Zameer, A., Muneeb, M.: Predictions for Covid-19 with deep learning models of LSTM, GRU and BI-LSTM. *Chaos, Solitons Fractals* **140**, 110212 (2020)
23. Trajanoska, M., Trajanov, R., Eftimov, T.: Dietary, comorbidity, and geo-economic data fusion for explainable Covid-19 mortality prediction. *Expert Syst. Appl.* **209**, 118377 (2022)
24. Wang, D., Zhang, S., Wang, L.: Deep epidemiological modeling by black-box knowledge distillation: an accurate deep learning model for covid-19. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 15424–15430 (2021)
25. Xiao, C., et al.: C-watcher: A framework for early detection of high-risk neighborhoods ahead of covid-19 outbreak. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4892–4900 (2021)
26. Xue, J., Yabe, T., Tsubouchi, K., Ma, J., Ukkusuri, S.: Multiwave Covid-19 prediction from social awareness using web search and mobility data. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4279–4289 (2022)
27. Zhang, P., Wang, Z., Huang, Y., Wang, M.: Dual-grained directional representation for infectious disease case prediction. *Knowl.-Based Syst.* **256**, 109806 (2022)
28. Zheng, S., Gao, Z., Cao, W., Bian, J., Liu, T.Y.: Hierst: a unified hierarchical spatial-temporal framework for Covid-19 trend forecasting. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 4383–4392 (2021)



Lead-Aware Hierarchical Transformer and Convolution Fusion Network for ECG Classification

Yuang Zhang¹, Binyu Wang¹, Liping Wang^{1(✉)}, and He Huang²

¹ East China Normal University, Shanghai, China

{51255902045, 51205902098}@stu.ecnu.edu.cn, lipingwang@sei.ecnu.edu.cn

² Shanghai Huayuan Chuangxin Software, Shanghai, China

yellowriver@ntesec.com.cn

Abstract. ECG classification is a typical and practical multivariate time series classification task. Recently, ECG classification with deep networks has been widely researched and achieved promising results. However, most of existing works focus on multi-class rather than multi-label ECG classification, while the latter is more clinically practical. Moreover, information within and between specific ECG leads can be further mined and utilized by deep learning models. Therefore, we develop Lead-aware Hierarchical Transformer and Convolution fusion Network (LHTC-Net). It integrates an Attention Convolution module and a Hierarchical Transformer module to extract both local and long-term dependency in ECG signals. In constructing the hierarchical transformer, we design three novel Window-based Transformer Blocks dedicated to local, global, and lead-specific information respectively. Additionally, a lead-aware mechanism is proposed to capture lead-specific information. Experiments show that LHTC-Net outperforms five SOTA methods with 82.67% and 78.53% micro F₁ scores on two real-world datasets. Extensive ablation studies demonstrate the roles of different modules of LHTC-Net, providing insights into our algorithm and ECG classification task.

Keywords: Time series · multi-label classification · ECG

1 Introduction

Heart disease is a major threat to human health and receives wide attention. According to WHO [21], it is the leading cause of human deaths worldwide. Electrocardiography (ECG) is a technique that uses an electrocardiogram machine to record electrical activity generated by each cardiac cycle of the heart, which is one of the most commonly used non-invasive, simple, and cost-effective methods of screening and diagnosing cardiac arrhythmias and cardiovascular diseases. The presentation of the same arrhythmia type can vary from patient to patient due to the complexity and diversity of ECG, making accurate diagnosis challenging for even experienced physicians. Awni Y Hannun et al. [4] counted the diagnostic

Y. Zhang, B. Wang—The first and second author contribute equally to this work.

results of several cardiologists on 328 single-lead ECG test data and their average F_1 score was only 0.78. Automated diagnosis could assist cardiologists in clinical practice and mitigate insufficiency of medical resources. Therefore, ECG classification is a crucial and challenging task.

Lots of research has been devoted to ECG classification over decades. Traditionally, ECG classification algorithms first segment the ECG signal into a number of heartbeats and extract hand-designed medical features for beat-level classification with an expert system or a statistical-based machine learning method like SVM. However, the effectiveness is limited, since extracting hand-designed features requires extensive medical domain knowledge and a complex pre-processing process. Therefore, the accuracy is significantly lower than that of cardiologists.

In recent years, deep learning has exhibited strong ability in many fields including computer vision, natural language processing and so on. It has also been widely used in the medical field, such as medical image segmentation, drug development, and genetic analysis. Deep neural networks can automatically extract features from large-scale data in an end-to-end manner without manual feature extraction, significantly improving the accuracy of ECG classification.

The most practical data type of ECG is a standard static ECG. It contains 12 leads (each as a time series) and lasts from a few to a dozen seconds. It is hard for traditional time series methods such as LSTM to capture its long-term dependencies across multiple heartbeats. To tackle this, many existing works split the whole ECG signal into heartbeats [9, 18, 26] or short segments with a fixed length [4, 16]. Recently, some works use residual convolutional network [25] or CNN with LSTM [19] or CNN with Transformer [30] to directly classify the whole ECG record. Most of them focus on multi-class rather than multi-label ECG classification. However, the latter is more suitable for clinical needs where several rhythm types may exhibit in the same record. Besides, different leads record heart activity from different spatial perspectives, reflecting heart condition comprehensively. A standard 12-lead ECG can observe more abnormalities than a single-lead ECG [8]. For example, the diagnosis of right ventricular hypertrophy is impossible with single-lead ECG, since it requires observation of abnormal waveforms in leads V1, V2, and V6.

Based on the above observations, our paper focuses on fully utilizing lead-specific information across multiple heartbeats of multi-lead ECGs. Our main contributions are summarized as follows:

- (1) We propose **L**ead-aware **H**ierarchical **T**ransformer and **C**onvolution fusion **N**etwork (LHTC-Net), a novel end-to-end deep learning model for multi-label classification on multi-lead ECG signals. LHTC-Net is mainly composed of two modules, Convolution and Transformer. The Attention Convolution module extracts local features in ECG signals, and the Hierarchical Transformer extracts long-term dependency across multiple heartbeats.
- (2) We propose a lead-aware mechanism by adopting self-attention with windows. The lead-aware mechanism enables the model to capture the lead-specific information, which helps to diagnose more arrhythmia types.

Table 1. Notations and descriptions

Notation	Description
m, n	number of leads, signal length
u	number of arrhythmia classes
$X \in \mathbb{R}^{m \times n}$	input ECG signal
$R \in \mathbb{R}^u, r_i$	ground-truth labels, and label of i th arrhythmia type
$\hat{R} \in \mathbb{R}^u, \hat{r}_i$	predicted labels, and predicted possibility of i th arrhythmia type
$Y^{(k)}, y^{(k)}$	output of k th stage of lead-aware hierarchical transformer module, and a patch of $Y^{(k)}$
$Y^{(0)} \in \mathbb{R}^{m \times \frac{n}{4} \times C}$	output of ECG patch embedding layer
$y^{(0)} \in \mathbb{R}^C$	a patch of $Y^{(0)}$
C	patch embedding feature dimension
L	number of patches of the current block in lead dimension
T	number of patches of the current block in temporal dimension
L'	window size of LW-MSA in lead dimension
T'	window size in TW-MSA in temporal dimension
h	number of attention heads in the current stage
$E \in \mathbb{R}^{m \times C}$	absolute positional encoding parameter for lead dimension
$B_i \in \mathbb{R}^{l \times l}$	relative position bias of i th attention head for temporal dimension
$Z^{(k)}$	output of k th stage of attention convolution module
$Y' \in \mathbb{R}^{8C}$	output of hierarchical transformer module after GeM pooling
$Z' \in \mathbb{R}^{256}$	output of attention convolution module after GeM pooling

- (3) We evaluate LHTC-Net with two real-world publicly available datasets, Hefei High-Tech Cup dataset [6] and Chapman dataset [33]. Experimental results show that our model achieves 82.67% and 78.53% micro F₁ scores on the two datasets respectively, outperforming five state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 introduces the problem definition. In Sect. 3, we introduce the details of the proposed model, LHTC-Net. The experiment results are shown in Sect. 4. Finally, we draw the conclusion in Sect. 5.

2 The Problem Statement

In this paper, we focus on multi-label classification problem of multi-lead ECG signals: taking a multi-lead ECG signal as input and output several arrhythmia labels. This task is also the most common clinical application for ECG.

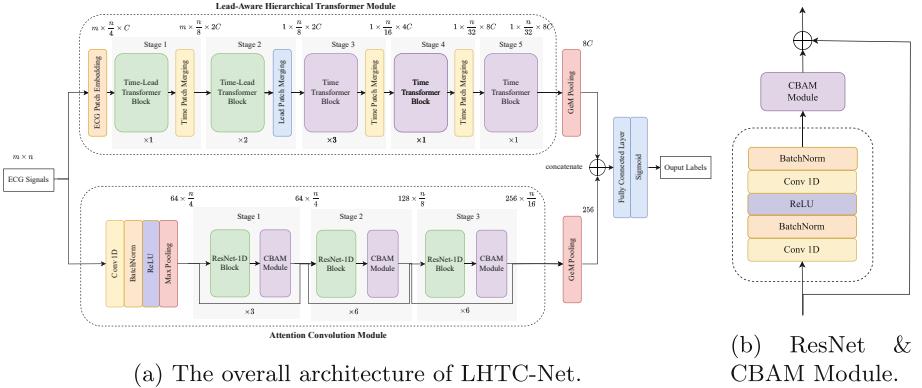


Fig. 1. Illustration of proposed LHTC-Net.

The problem can be formally described as: Given a multi-lead ECG signal $X \in \mathbb{R}^{m \times n}$ as input, which can be described as the following formula:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \cdots & x_{mn} \end{bmatrix} \quad (1)$$

where n is the length of a single signal and m is the number of leads. Each X has a sequence of ground-truth labels $R \in \mathbb{R}^u$ as (2). Multiple r_i can be 1, representing the existence of i th arrhythmia type. The model predicts by outputting $\hat{R} \in \mathbb{R}^u$ with each \hat{r}_i the predicted possibility of the corresponding arrhythmia.

$$R = [r_1 \ r_2 \ \cdots \ r_u], r_i \in \{0, 1\} \quad (2)$$

3 Method

3.1 Overall Architecture

Fig. 1a shows the overview of proposed **L**ead-aware **H**ierarchical **T**ransformer and **C**onvolution fusion **N**etwork (LHTC-Net). LHTC-Net consists of three main parts: a lead-aware hierarchical transformer module, an attention convolution module, and a classification head. Given a multi-lead ECG signal $X \in \mathbb{R}^{m \times n}$, transformer module and convolution module first extract features $Y^{(5)} \in \mathbb{R}^{1 \times \frac{n}{32} \times 8C}$ and $Z^{(3)} \in \mathbb{R}^{256 \times \frac{n}{16}}$ respectively, where C is the dimensionality of patch embedding feature. The transformer module can capture long-term dependency and extract lead information with lead-aware mechanism. The attention convolution module focuses on local features without lead information. Then GeM pooling [23] reduces the features $Y^{(5)}$, $Z^{(3)}$ to $Y' \in \mathbb{R}^{8C}$, $Z' \in \mathbb{R}^{256}$. Following, we introduce the details and mechanisms on a module-wise scale.

3.2 Lead-Aware Hierarchical Transformer

Since traditional convolution-based method cannot directly capture features on global scale, we designed lead-aware hierarchical transformer inspired by Swin Transformer [17]. Transformer [28] is an attention-based method, it performs better than CNN and LSTM on long dependency. In Swin Transformer, a hierarchical transformer architecture designed for computer vision make the model adaptable to various scales and has linear computational complexity with respect to input image size using a shifted window-based attention mechanism.

Our lead-aware hierarchical transformer not only follows the hierarchical transformer architecture but also designs a lead-aware mechanism to capture features on various temporal scales as well as pay attention to crucial information in the lead dimension. An ECG patch embedding layer first splits the raw signal $X \in \mathbb{R}^{m \times n}$ into $m \times \frac{n}{4}$ non-overlapping patches and embeds each patch into $y^{(0)} \in \mathbb{R}^C$. The overall architecture is illustrated in Fig. 1a.

ECG Patch Embedding. Raw signal sequences cannot be input directly due to their excessive lengths. Therefore, for a given input $X \in \mathbb{R}^{m \times n}$, we firstly split it into non-overlapping patches for each lead, where each patch includes 4 sample points, gaining $m \times \frac{n}{4}$ patches. Figure 2(a) shows an example when $m = 4$ and $\frac{n}{4} = 16$. Then a linear layer followed by layer normalization (LN) [1] project the features of each patch into a C dimension feature vector $y^{(0)} \in \mathbb{R}^C$.

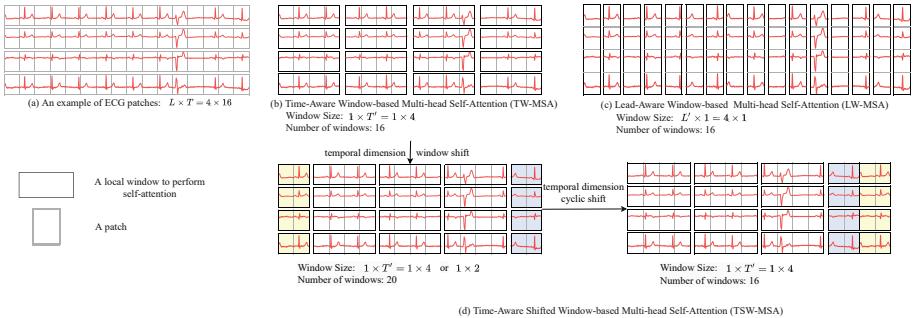


Fig. 2. An illustrated example of window-based self-attention. (a) Splitting ECG patches, (b) TW-MSA, (c) LW-MSA, (d) TSW-MSA.

Window-Based Transformer Blocks. We adapt Swin Transformer blocks by replacing the multi-head self-attention (MSA) in vanilla Transformer blocks with **Time-aware Window-based MSA** (TW-MSA), **Lead-aware Window-based MSA** (LW-MSA) and **Time-aware Shifted Window-based MSA** (TSW-MSA) as illustrated in Fig. 3 to develop a time-lead transformer block and a time transformer block. GELU activation [7] is used in MLP layers.

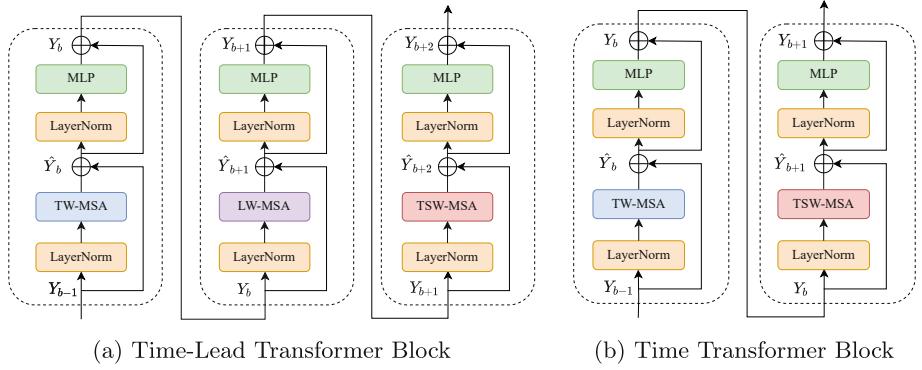


Fig. 3. Architectures of two different window-based Transformer Blocks.

Window-Based MSA: TW-MSA and LW-MSA. Since global MSA on ECG patches generates enormous computation and memory costs, we follow Swin Transformer to calculate MSA within non-overlapping windows. Furthermore, we introduce an ECG-specific inductive bias: cardiologists usually focus on single-lead waveform over time and the combination of different leads at one same time point. Therefore, we choose to use 1D windows on temporal and lead dimensions instead of a 2D window. Given $L \times T$ patches, window sizes $1 \times T'$ in TW-MSA and $L' \times 1$ in LW-MSA, the input patches are partitioned into non-overlapping windows $L \times \lceil \frac{T}{T'} \rceil$ and $1 \times T$. An example is shown in Fig. 2(a)(b)(c).

Shifted Window-Based MSA: TSW-MSA. Since MSA is calculated *within* each non-overlapping window, window-based MSA lacks connections *across* windows, ignoring global dependencies. To address this, we simplify the 2D spatial shifted window mechanism in Swin Transformer to a 1D temporal shifted window mechanism to extract features across temporal windows. Given $L \times T$ patches and a 1D window size of $1 \times T'$, which generate $L \times \lceil \frac{T}{T'} \rceil$ non-overlapping 1D windows, we then shift the window along the temporal dimension by $\lfloor \frac{T'}{2} \rfloor$ patches. An example is illustrated in Fig. 2(d). Similar to [17], cyclic shifting is used to keep all window sizes the same.

Time and Lead Patch Merging. To produce a hierarchical representation, we use the time patch merging layer in stages 1, 3, and 4 in Fig. 1a as well as lead merging layer in stage 2 to integrate information across all leads.

Time Patch Merging. In time patch merging layer, we concatenate the features of two temporal neighboring patches in the same lead and apply an LN layer. The temporal dimensionality halves and the feature dimensionality doubles.

Lead Patch Merging. In lead patch merging layer, we concatenate the feature of all leads at the same time point and then apply a linear layer followed by an

LN layer to keep the feature dimensionality consistent. The lead dimensionality is reduced to one and the feature dimensionality remains the same.

Hybrid Positional Encoding. As self-attention itself lacks positional information, we add a hybrid positional encoding to patch embedding and self-attention calculating. We use a learnable absolute positional encoding like [3, 14] for lead dimension and a learnable relative position bias like [10, 17, 24] for temporal dimension.

To implement absolute positional encoding for lead dimension, we add a trainable parameter $E \in \mathbb{R}^{m \times C}$ to the output patches of ECG patch embedding layer $Y^{(0)} \in \mathbb{R}^{m \times \frac{n}{4} \times C}$ as

$$\hat{y}_{ij}^{(0)} = y_{ij}^{(0)} + e_i \quad (3)$$

where $\hat{y}_{ij}^{(0)} \in \mathbb{R}^C$ and $y_{ij}^{(0)} \in \mathbb{R}^C$ are the input and output feature vector of j th time patch of i th lead in absolute positional encoding, $e_i \in \mathbb{R}^C$ is the positional encoding parameter for the i th lead.

For temporal dimension, a relative position bias $B_i \in \mathbb{R}^{l \times l}$ is used in TW-MSA and TSW-MSA by (4).

$$H_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}} + B_i\right) V_i \quad (4)$$

B_i is a learnable parameter representing the relative position along the temporal dimension in the range $[-l + 1, l - 1]$.

3.3 Attention Convolution Module

Convolution-based methods are widely used in many state-of-the-art ECG classification methods such as [4, 16, 32] and obtain outstanding performances. Our attention convolution module's architecture is similar to ResNet [5] with Convolutional Block Attention Module (CBAM) [29] in computer vision field. CBAM increases representation power by emphasizing meaningful features along both channel and spatial axes in an image. As the ECG signal is a kind of time-series data, we replace all the 2D convolutional layers in original CBAM with 1D convolutional layer and calculate the spatial attention map on 1D temporal dimension instead of 2D spatial dimension.

For the given ECG signals $X \in \mathbb{R}^{m \times n}$, we first use a 1D convolutional layer with kernel size equal to 15 and stride equal to 2 to downsample the input to a shape of $64 \times \frac{n}{2}$. A BatchNorm (BN) layer and ReLU activation are applied afterward, followed by a max-pooling layer with kernel size equal to 3 and stride equal to 2, further reducing the input to $Z^{(0)} \in \mathbb{R}^{64 \times \frac{n}{4}}$. There are 3 stages with 3, 6, and 6 stacked ResNet-1D blocks with CBAM respectively. As shown in Fig. 1b, there are two 1D convolutional layers with kernel size equal to 7, a ReLU activation, two BatchNorm layers, a CBAM module, and a shortcut connection in each block. At the beginning of stage 2 and stage 3, we add a

1D convolutional layer with a kernel size equal to 1 to halve the size of feature map and double the number of channels. Finally, the output of the last stage is $Z^{(3)} \in \mathbb{R}^{256 \times \frac{n}{16}}$.

3.4 Feature Fusion and Loss Function

After extracting features by transformer and convolution module, we obtain two features $Y^{(5)} \in \mathbb{R}^{1 \times \frac{n}{32} \times 8C}$ and $Z^{(3)} \in \mathbb{R}^{256 \times \frac{n}{16}}$. To integrate these two, we first reduce them to $Y' \in \mathbb{R}^{8C}$, $Z' \in \mathbb{R}^{256}$ by applying Generalized mean (GeM) pooling [23] and an LN layer. Then we concatenate Y' and Z' into a fusion feature. Finally, a fully connected layer with sigmoid function is applied for classification.

Suppose that we pool along the last dimension of a feature $S \in \mathbb{R}^{l' \times l''}$, GeM pooling is calculated as

$$g_i = \left(\frac{1}{l''} \sum_j^{l''} s_{ij}^p \right)^{\frac{1}{p}} \quad (5)$$

where g_i is the output of i th value and p is the learnable generalized mean power parameter. When p is 1, GeM pooling is same as average-pooling; when p is close to 0, it behaves like max-pooling.

Weighted binary cross-entropy loss (WBCE) is chosen as the loss function. WBCE can mitigate class imbalance problem by assigning a weight for each class according to its volume. Given an output $\hat{R} \in \mathbb{R}^u$ and the label $R \in \mathbb{R}^u$,

$$w_i = \frac{1}{\log(\text{count}(i) + 1)} \quad (6)$$

$$\text{Loss} = - \sum_i^u w_i (r_i \log(\hat{r}_i) + (1 - r_i) \log(1 - \hat{r}_i)) \quad (7)$$

where u denotes the number of classes, $\text{count}(i)$ denotes the number of records belonging to i th class in the training dataset.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our model on two real-world multi-lead multi-label ECG datasets, i.e., **Hefei High-Tech Cup** dataset [6] and **Chapman** dataset [33].

Hefei High-Tech Cup dataset [6]: contains 24,106 training records and 8,036 testing records. Each record is a labeled 8-lead ECG signal, sampled at 500 Hz and lasting for 10s. The 8 leads are I, II, V1, V2, V3, V4, V5, and V6. Each label contains 55 different classes of arrhythmia. We divide the 24,106 records for training into 19,283 records as the actual training set and 4,823 records as the validation set by stratified sampling based on labels.

Chapman Dataset [33]: contains 10,629 labeled records of standard 12-lead ECG signal, collected by Chapman University and Shaoxing Peoples Hospital, sampled at 500 Hz and lasting for 10 s. We focus on 29 arrhythmia labels with records more than 90, which are divided into 7,654 records as training set, 1,914 records as validation set, and 1,061 records as test set by stratified sampling based on labels.

Pre-processing. Baseline wandering is a common low-frequency noise existing in ECG signals. We remove it by using high-pass filtering to allow only signals above 0.5 Hz to pass. Then the signal is resampled to 2,048 sampling points per lead by the Fourier method [20]. To mitigate the impact of high-frequency noise, we apply data augmentation by randomly adding Gaussian additive noise to the input signals while training.

Implementation Details. The signal length n is set to 2,048, and lead numbers are 8 and 12 for Hefei High-Tech Cup and Chapman datasets, respectively. The embedding dimensionality in ECG patch embedding layer is $C = 96$. We fix the dimensionality of each attention header to be 8, so the number of attention heads in the five stages is [12, 24, 24, 48, 96] in order. The window size in TW-MSA is set to $1 \times T' = 1 \times 32$, and the window size L' in LW-MSA is equal to L . To mitigate overfitting, dropout [27] is applied in ResNet-1D block after ReLU activation with drop ratio 0.3, and stochastic depth [13] is applied in both transformer and convolutional blocks with a ratio of 0.4.

All experiments are implemented in PyTorch and trained with a system with 128 GB RAM, 28 Intel Xeon Gold 6258R 2.70 GHz CPUs, and Nvidia Tesla A100 GPU. Our model was optimized using Adam [15] with a learning rate of 0.0003. Learning rate decay happens by a factor of 10 at epoch 16, 32, 64, and 80. The batch size is 128 for Hefei High-Tech Cup dataset and 96 for Chapman dataset. We train the model for 128 epoch iterations and choose the model with the best micro- F_1 score in the validation set as the final model. Finally, the classification result for each label is determined by a threshold of 0.5.

Evaluation Metric. In our research, recall, precision, and F_1 score are used to measure the performance of models on each class, and we choose micro-recall, micro-precision, and micro- F_1 as the metrics on all classes. F_1 score is the harmonic mean of the precision and recall, which is a metric representing the overall performance.

4.2 Results on Hefei High-Tech Cup Dataset

We evaluate LHTC-Net on Hefei High-Tech Cup dataset and report the result of each class and the overall performance in Fig. 4. The arrhythmia labels are ordered by the data quantity. Due to class imbalance problem, the first 10 arrhythmia labels cover 98.69% data records and 60.52% positive labels. In Fig. 4,

we observe: (1) Generally, the model performs better on classes with more data. On classes such as right atrial enlargement with only 31 records and complete left bundle branch block with only 23 records, F_1 score drops to 0. Although we use weighted binary cross-entropy loss to alleviate class imbalance problem, data quantity still significantly affects the performance. (2) On classes with larger data volumes, recall and precision are more balanced, thus their F_1 scores are relatively high. On classes with smaller data volumes, the precision may be high but the recall is very low, indicating that the model tends to neglect to diagnose these arrhythmia classes.

4.3 Comparison With Known Methods

Baselines. We compare LHTC-Net with the following five end-to-end ECG classification models:

16-layer CNN [31]: A 1D-CNN based method with 16 layers for 10s ECG classification.

ECGNet [19]: A combination network of multi-scale 1D-CNN and LSTM for arrhythmia classification.

DDNN [2]: A DenseNet-based [12] network with SE module [11] for atrial fibrillation detection on 12-lead ECG records.

ResNet-1D [25]: A ResNet-based [5] 1D-CNN network for automatic diagnosis of 12-lead ECG.

STCT [22]: A Conv-Transformer network which extracts spatial and temporal information from ECG signals by using 1D-CNN and CvT for cardiac arrhythmias recognition, where CvT [30] is a convolutional adaption of Vision Transformer.

Same loss function as our method is used on all baseline models to fit the multi-label ECG classification problem.

Results. As shown in Table 2, we compare LHTC-Net with the five baselines on the two datasets. The best and second-best results on each evaluation metric are in bold and underlined, respectively. LHTC-Net reaches the highest micro- F_1 scores of 82.67% and 78.53% on the two datasets. LHTC-Net can better balance recall and precision and obtain the best overall performance in both datasets. In these baselines, two pure CNN models of 16-layer CNN and DDNN perform worst, and ResNet-1D is much better with its residual connections. The CNN and RNN combined model, ECGNet, obtains better performance than vanilla CNN models by extracting temporal features using LSTM. STCT is also a transformer and convolution fusion model with very high precision, but its overall performance is hindered due to its low recall.

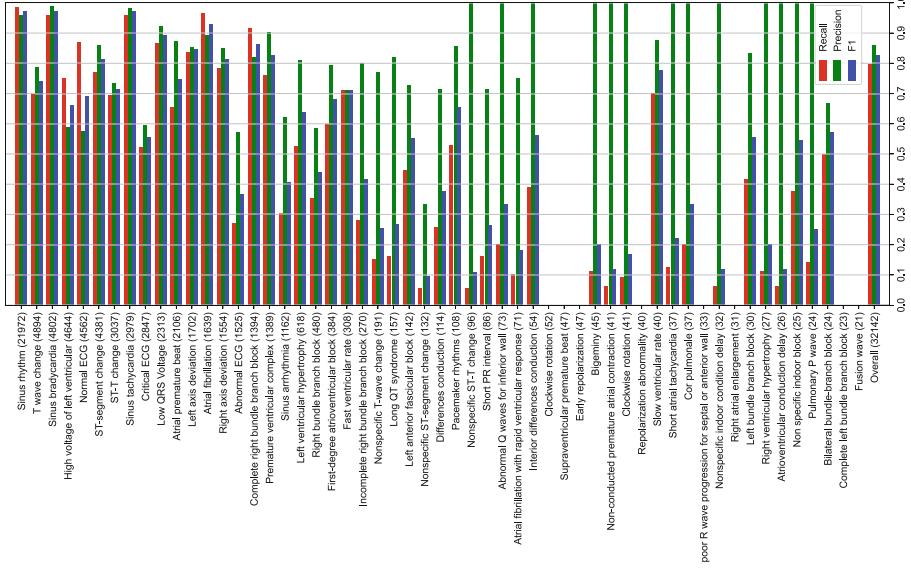


Fig. 4. Results on Hefei High-Tech Cup Dataset. This figure shows the recall, precision, F1 score on each class, and their micro versions on the overall test dataset. The arrhythmia labels are ordered by the data quantity of the whole dataset, which is marked in brackets.

Table 2. Comparision of the Results From the Proposed Method With Known Methods.

Dataset	Hefei High-Tech Cup			Chapman				
	Method	Recall	Precision	F1 score	Method	Recall	Precision	F1 score
16-layer CNN [31]	73.76%	83.67%	78.41%	64.53%	82.27%	72.33%		
ECGNet [19]	77.70%	85.80%	81.55%	69.23%	84.43%	76.08%		
DDNN [2]	69.62%	82.77%	75.62%	62.15%	82.84%	71.02%		
ResNet-1D [25]	79.84%	83.37%	81.57%	<u>72.44%</u>	83.39%	<u>77.53%</u>		
STCT [22]	77.50%	86.94%	81.95%	69.75%	85.33%	76.76%		
LHTC-Net	79.52%	86.08%	82.67%	74.04%	83.60%	78.53%		

4.4 Ablation Study

To verify the effectiveness of important components in LHTC-Net, we evaluate several variants of LHTC-Net on the two datasets.

Transformer and Convolution Module. The results using only Lead-Aware Hierarchical Transformer Module or Attention Convolution Module separately are shown in Table 3. First, both modules individually underperform LHTC-Net. Second, the convolution module outperforms the transformer module in all

metrics on both datasets. These observations suggest that for multi-lead ECG classification tasks, local features extracted by convolution are more important. However, the transformer module we designed extracts global and lead information to complement the local features, further improving the model.

Table 3. Ablation Study on Transformer and Convolution Module.

Dataset	Hefei High-Tech Cup			Chapman		
Metric	Recall	Precision	F_1 score	Recall	Precision	F_1 score
Convolution only	80.63%	84.16%	82.36%	72.85%	84.52%	78.26%
Transformer only	78.98%	83.68%	81.26%	72.03%	83.06%	77.15%
LHTC-Net	79.52%	86.08%	82.67%	74.04%	83.60%	78.53%

Lead-Aware Mechanism. To verify the effectiveness of the lead-aware mechanism, we modify our models as follows: First, we split non-overlapping patches of $m \times 4$ sample points instead of 1×4 sample points in the ECG patch embedding layer, so there are $1 \times \frac{n}{4}$ patches after splitting. Second, we remove all LW-MSA sub-blocks and the lead patch merging layer. As shown in Table. 4, the modified model performs significantly worse than LHTC-Net. The above results indicate that the lead information captured by the lead-aware mechanism can enhance the performance of LHTC-Net.

Shifted Window Mechanism. To verify the effectiveness of the shifted window mechanism, we replace all TSW-MSA sub-blocks with TW-MSA sub-blocks. As shown in Table. 5, shifted window mechanism benefits LHTC-Net by 0.37% and 0.43% F_1 score on the two datasets, respectively. This mechanism exchanges information between adjacent temporal windows and indirectly establishes the global attention mechanism, which effectively enhances the performance.

Table 4. Ablation Study on Lead-Aware Mechanism.

Dataset	Hefei High-Tech Cup			Chapman		
Method	Recall	Precision	F_1 score	Recall	Precision	F_1 score
w/o Lead-Aware	78.77%	86.23%	82.33%	73.27%	83.45%	78.03%
LHTC-Net	79.52%	86.08%	82.67%	74.04%	83.60%	78.53%

Table 5. Ablation Study on Shifted Window Mechanism.

Dataset	Hefei High-Tech Cup			Chapman		
Method	Recall	Precision	F_1 score	Recall	Precision	F_1 score
w/o Shifted Window	79.06%	85.81%	82.30%	73.47%	83.34%	78.10%
LHTC-Net	79.52%	86.08%	82.67%	74.04%	83.60%	78.53%

Positional Encoding(PE). Table 6 shows comparisons of different PEs on both lead and temporal dimensions. We investigate the effect of PEs by fixing PE on one dimension and changing the other. Fixing the temporal PE as relative positional bias, we find that absolute PE is better for lead dimension, probably because each lead has its physical meaning, which can be directly encoded by absolute PE. In contrast, the time interval length between two patches can be better represented by relative PE.

Table 6. Ablation Study on Positional Encoding.

Positional Encoding		Hefei High-Tech Cup			Chapman		
Lead	Temporal	Recall	Precision	F_1 score	Recall	Precision	F_1 score
-	relative	79.15%	86.26%	82.56%	72.91%	84.33%	78.20%
relative	relative	79.27%	86.12%	82.55%	72.60%	83.92%	77.85%
absolute	relative	79.52%	86.08%	82.67%	74.04%	83.60%	78.53%
absolute	absolute	78.99%	86.40%	82.61%	73.32%	83.86%	78.23%
absolute	-	79.64%	85.81%	82.61%	73.27%	84.59%	78.52%

4.5 Discussion

A visualized case study illustrates proposed lead-aware mechanism in Fig. 5, showing a test record from Hefei High-Tech Cup dataset. The outputs by different variants of LHTC-Net of this record is in Table 7. The label is *Sinus rhythm*, *T wave change*, and LHTC-Net outputs correctly. However, variants without lead-aware mechanism in transformer module ignore the T wave change and output *Normal ECG* instead, indicating the important role of lead-aware mechanism in this case. To explain this, we can observe from Fig. 5 that T wave change only happens on Lead V3. In marked areas, normal T waves change into Camel Hump T waves. However, T waves on other leads are normal. Therefore, the model needs to accurately capture the information about the changes in ECG signals in different leads. Lead-aware mechanism is exactly able to focus on the important features in different leads with LW-MSA, thus sharply detecting the change of T wave in V3 leads in Fig. 5.

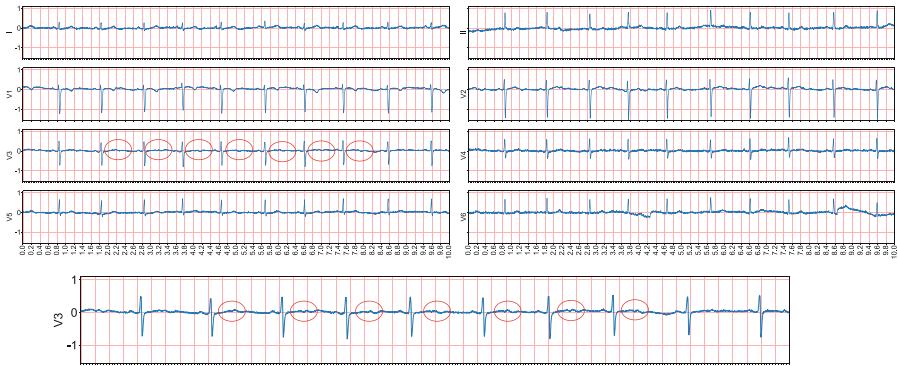


Fig. 5. An illustrated case example. The location of the T wave change is marked with a red circle. (Color figure online)

Table 7. Outputs of Different Variants of LHTC-Net on Fig. 5.

Method or Label	Outputs or Labels
Label	Sinus rhythm, T wave change
LHTC-Net	Sinus rhythm, T wave change
w/o Lead-Aware	Sinus rhythm, Normal ECG
Convolution only	Sinus rhythm, Normal ECG
Transformer only	Sinus rhythm, T wave change

5 Conclusion

In this paper, we propose LHTC-Net, an end-to-end deep learning network fused with transformer and convolution, designed for multi-lead ECG classification. We develop a window-based self-attention mechanism to utilize both temporal and lead-specific information for a better diagnosis of arrhythmias. The structure of transformer and convolutional fusion networks allows the model to consider both local waveform features and long-range dependencies across multiple heartbeats. Experiments on two real datasets verify the effectiveness of our LHTC-Net, which outperforms state-of-the-art and achieves 82.67% micro F₁ score on Hefei High-Tech Cup dataset and 78.53% micro F₁ score on Chapman dataset.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016)
2. Cai, W., et al.: Accurate detection of atrial fibrillation from 12-lead ECG using deep neural network. *Comput. Biol. Med.* **116**, 103378 (2020)
3. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: *ICLR* (2021)

4. Hannun, A.Y., et al.: Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat. Med.* **25**(1), 65–69 (2019)
5. 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)
6. 2019 tianchi hefei high-tech cup ecg human-machine intelligence competition. <https://tianchi.aliyun.com/competition/entrance/231754/introduction>. Accessed Jan 2024
7. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus) (2023)
8. Hong, S., Zhou, Y., Shang, J., Xiao, C., Sun, J.: Opportunities and challenges of deep learning methods for electrocardiogram data: systematic review. *Comput. Biol. Med.* **122**, 103801 (2020)
9. Hou, B., Yang, J., Wang, P., Yan, R.: LSTM-based auto-encoder model for ECG arrhythmias classification. *IEEE Trans. Instrum. Meas.* **69**(4), 1232–1240 (2019)
10. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3588–3597 (2018)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141 (2018)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708 (2017)
13. 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.) *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV*, pp. 646–661. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_39
14. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT*, pp. 4171–4186 (2019)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations*, In: *ICLR* (2015)
16. Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patient-specific ecg classification by 1-D convolutional neural networks. *IEEE Trans. Biomed. Eng.* **63**(3), 664–675 (2015)
17. Liu, Z., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022 (2021)
18. Mousavi, S., Afghah, F.: Inter-and intra-patient ecg heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1308–1312. IEEE (2019)
19. Murugesan, B., Ravichandran, V., Ram, K., Preejith, S., Joseph, J., Shankaranarayana, S.M., Sivaprakasam, M.: Ecgnnet: deep network for arrhythmia classification. In: *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–6. IEEE (2018)
20. Nussbaumer, H.J.: The fast Fourier transform. In: *Fast Fourier Transform and Convolution Algorithms*, pp. 80–111. Springer (1981). https://doi.org/10.1007/978-3-662-00551-4_4

21. Organization, W.H., et al.: World Health Statistics Overview 2019: Monitoring Health For The SDGS, Sustainable Development goals. World Health Organization, Tech. rep. (2019)
22. Qiu, Y., Chen, W., Yue, L., Xu, M., Zhu, B.: STCT: spatial-temporal conv-transformer network for cardiac arrhythmias recognition. In: Li, B., Yue, L., Jiang, J., Chen, W., Li, X., Long, G., Fang, F., Yu, H. (eds.) Advanced Data Mining and Applications: 17th International Conference, ADMA 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings, Part I, pp. 86–100. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-030-95405-5_7
23. Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(7), 1655–1668 (2018)
24. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
25. Ribeiro, A.H., et al.: Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nat. Commun.* **11**(1), 1–9 (2020)
26. Saadatnejad, S., Oveisi, M., Hashemi, M.: LSTM-based ECG classification for continuous monitoring on personal wearable devices. *IEEE J. Biomed. Health Inform.* **24**(2), 515–523 (2019)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
28. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems* **30** (2017)
29. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: convolutional block attention module. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (9 2018)
30. Wu, H., et al.: Cvt: introducing convolutions to vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31 (2021)
31. Yıldırım, Ö., Pławiak, P., Tan, R.S., Acharya, U.R.: Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Comput. Biol. Med.* **102**, 411–420 (2018)
32. Zhang, D., Yang, S., Yuan, X., Zhang, P.: Interpretable deep learning for automatic diagnosis of 12-lead electrocardiogram. *Iscience* **24**(4), 102373 (2021)
33. Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., Rakovski, C.: A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Sci. Data* **7**(1), 1–8 (2020)



Reinforcement Learning from Clip

Shaoqiang Zhu[✉], Kejia Zhang^(✉), and Haiwei Pan

Harbin Engineering University, No. 145, Nantong Street, Nangang District, Harbin,
Heilongjiang, China
`{chuckiezhu,kejiazhang,panhaiwei}@hrbeu.edu.cn`

Abstract. Reinforcement Learning (RL) algorithms have faced unstable studying processes, slow learning speed, and sparse rewards. Many studies provide solutions to those problems in both online and offline learning methods. However, these methods require either large amounts of offline data or interactive computation. This paper presents a RL framework: Reinforcement Learning From Clip (RLFC), which can learn from clips and be viewed as a plug-in and easily integrated into current RL frameworks. Using a small amount of additional information, without adding a lot of online calculations, or increasing the number of interaction rounds, this plug-in gives the speed and stability in the early stage of reinforcement learning huge improvements. In RLFC, one needs to use a scorer pre-trained on quantitative clips as an expert evaluator, this scorer gives a score as an expert. When the agent acts, the scorer gives a score based on the action. Due to the existence of the scorer, the problem of sparse reward is alleviated, thus making the training process stable and having a faster training speed. Experiments show that after integrating RLFC, mainstream algorithm frameworks such as DQN, PPO, and SAC algorithms all get more stable and faster.

Keywords: Reinforcement Learning · Learning From Expert Clip · RL plug-in · enhance stability · reward sparse

1 Introduction

Reinforcement learning(RL) is a general framework in which an agent facing a decision-making problem makes autonomous decisions to interact with and learn from the environment [2]. It has recently been developed into deep reinforcement learning [9], which combines neural networks and RL to enable this kind of agent to learn interactively in complex environments. RL methods have been applied in different scenarios in real-world domains, such as playing video games [10], playing chess [15], making animation demonstrations [12], and automatic Driving [1].

In existing methods, the tasks to be learned by the agent are generally divided into two categories based on the way the agent takes actions: discrete action control tasks and continuous action control tasks. All tasks can be regarded as decision-making problems. In one task, the agent takes the action (that the agent

thinks best) based on the state of the environment it observes. After that, the environment changes based on the action taken by the agent and renders rewards to the agent.

For these two different tasks, existing methods are generally based on two basic reinforcement learning algorithms: Deep Q Network (DQN) [9] and Actor Critic Algorithm (AC) [3]. However, due to the characteristics of real-world tasks, both methods face common problems: poor stability of the learning process, slow learning speed, and sparse reward problems.

This paper proposes a plug-in algorithm called Reinforcement Learning From Clip (RLFC). This algorithm uses a small amount of offline expert data for reinforcement learning, which can alleviate the above three problems. Unlike the typical two-part structure RL, this framework introduces a third part: the scorer.

For example, in the classic environment of Cliff Walking, considering a sparse reward scenario, the agent must arrive at the End Point as quickly as possible while protecting himself from falling on the cliff as shown in Fig. 1. Given some random trajectory clips, the green points stand for traditional rewards for each step, and the red points are rewards from a well-trained scorer, the actual rewards the agent receives are the sum of the red and the green. Cause any network has errors, this scorer is designed to offer rewards between zero and one. Even if this scorer gives some familiar rollouts a 0, the agent’s learning process will not be changed in the worst case.

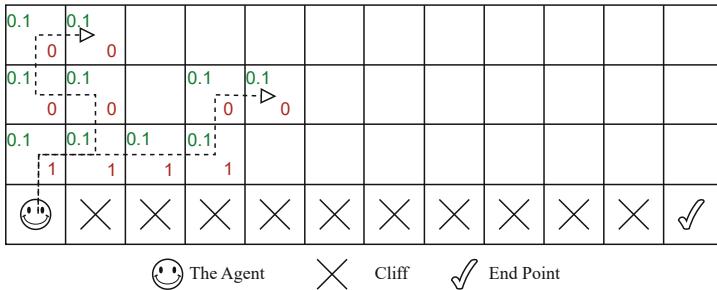


Fig. 1. We show the original rewards (left top, green) and scorer rewards (right bottom, red) of two random trajectory clips. In a sparse reward scenario, the agent gets 0.1 points at every step, he cannot know the right path till he completes the right path, which is so contradictory that he must waste much time to get the one in the training early stage. With a scorer, who is trained by small amounts of expert demonstrations, the agent gets an extra 1 point at every step if this step is similar to the expert otherwise 0. Intuitively, the scorer can alleviate the sparse reward problem and accelerate the speed of finding the right path in the early training stage. (Color figure online)

The main contributions of this paper can be summarized as:

- To our best knowledge, this paper proposes for the first time to use some offline experience fragments to pre-train a scorer to guide the agent during

the training process, accelerate the agent learning process, and enhance the stability of learning.

- This paper proposes a new definition of the scorer and proposes that at least a scorer exists in any Markov Decision Process(MDP) with a solution, specifying the application scope of the plug-in proposed in this paper. The proof is given in Appendix A.
- We used fragments of the Datasets for Deep Data-Driven Reinforcement Learning (D4RL) expert datasets to pre-train the scorer, compared multiple algorithms in multiple environments in the gym environment, and conducted experiments in discrete and continuous action spaces. Experiments show that the methods proposed in this paper can accelerate the learning process in the early stage and greatly improve the algorithm's stability.

2 Related Works

Deep Q Network(DQN) was proposed by [9], which uses a deep learning network combined with RL methods to solve control problems in high-dimensional space. On this basis, there have been many improvements, such as using two Q networks to update each other [17], solving the problem of overestimation of DQN value estimation [6]; [18] adopted a novel network structure, the state-action value in the network structure is decomposed into a state value function and an advantage function, that is $Q(s, a) = V(s) + A(s, a)$, thereby better fitting the reward function.

Actor-Critic(AC) is a classic RL algorithm that solves control problems in continuous action space and model-free situations. [3] proposed the Actor-Critic network reinforcement learning model, which greatly improved RL learning efficiency in continuous action spaces. On this basis, [13] and [14] developed the Trust Region Policy Optimization(TRPO) and Proximal Policy Optimization(PPO) algorithms, which are more effective and perform better than the AC algorithm.

Imitation Learning (IL) is a method of learning behaviors from offline demonstrations. These demonstrations are generally expert behaviors and contain information about states, actions, rewards, and next states. Since there is no need to interact with the environment, it can learn at an extremely fast speed. Imitation learning has two major directions: behavioral cloning (BC) and inverse reinforcement learning (IRL). The BC method, such as [16], treats the interaction between the agent and the environment as supervised learning, allowing the agent to learn to take specific actions in a specific type of environment; The IRL method [11] mainly infers the actual reward function through expert demonstrations and then uses this reward function to guide the behavior of the agent. Recently, a kind of imitation learning has developed imitation learning again [12], which eliminates the action information required for imitation learning, simplifies the data collection process and only needs to collect short videos from online video platforms (such as YouTube).

Previous works have revealed common problems in reinforcement learning: 1. sparse reward problem; 2. slow learning speed problem; 3. unstable learning speed problem. Although the methods proposed by [17] and [18] alleviate the problem of inaccurate reward function in the early stage of training by changing the structure of the reward function, thereby significantly speeding up the training process, they are still difficult to deal with in environments with sparse rewards. In the IL scenario, the agent directly clones the expert's actions through supervised learning, but the performance of the agent in a real-world application is proportional to the coverage of the state-action pairs in the expert's demonstration, furthermore, collecting the data is expensive.

3 Preliminary

In this section, we introduce notations and background in the first place, then give some definitions and proofs.

Notations and Background. In the RL framework, A Markov Decision Process(MDP) is expressed as $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$, where \mathcal{S} and \mathcal{A} are used to represent the state space and the action space respectively; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ represents the reward; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P}(\mathcal{S})$ represents state transition. $\gamma \in [0, 1]$ is the discount factor. At time step t , the state transition can be written as $\mathcal{T}(s'|s, a) = Pr(s')$. Reward, given by the environment, is written as $R(s, a, s') = E[r_t | s_t = s, a_t = a, s_{t+1} = s']$. A RLFC framework defines MDP as $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma, m)$, $m \in [0, 1]$ is score given by a scorer $\phi : s_1 \times s_2 \times \dots \times s_k \rightarrow m$; This paper uses \mathcal{H} to represent a state sequence with len k (the first $k - 1$ are historical states, and the k th is the future state) and as the input to a scorer, the ϕ is defined as $\phi : \mathcal{H} \rightarrow [0, 1]$. Moreover, there is a scorer function ϕ maps m to reward from the scorer: $r_t^s = \psi(m)$. The agent's goal is now to learn an optimal policy π^* under a scorer ϕ and a scorer function ψ so that this optimal policy maximizes cumulative return, that is:

$$\max G = \max E_{s_0, a_t \sim \pi(\cdot | s_t), s_{t+1} \sim \mathcal{T}(s_t, a_t)} \left[\sum_{t=0}^n (\gamma^t (\mathcal{R}(s_t, a_t) + \psi(\phi(\mathcal{H}))) \right] \quad (1)$$

Now we introduce the definition of a scorer:

Definition 1 (Scorer). *We define that: Given a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ with at least one expert solution, a scorer ϕ is a function, for any subsequence $\mathcal{H}_k = < s_{t-k+2}, \dots, s_t, s_{t+1} >$ of a sequence $\mathcal{H} = < s_1, s_2, \dots, s_n >$ generated in \mathcal{M} by any policy π , the scorer can give a score $m (m \in [0, 1])$ to identify the matching degree between \mathcal{H}_k and expert behavior.*

For the existence of a scorer, we give the following theorem (The proof can be founded at Appendix A):

Theorem 1. *Given a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ with at least one human expert solution, There exists at least one scorer ϕ .*

4 Methodology

In this section, we first describe the whole framework, then introduce the idea of training the scorer, and then the idea of training the agent with the scorer.

In this RLFC framework, an extra part scorer is introduced to the standard RL framework, this scorer gives a score m depending on $k - 1$ historical states and 1 future state, then a function ψ gives a score reward based on m , and the agent gets a reward from the environment and a score reward from scorer. As shown in Fig. 2.

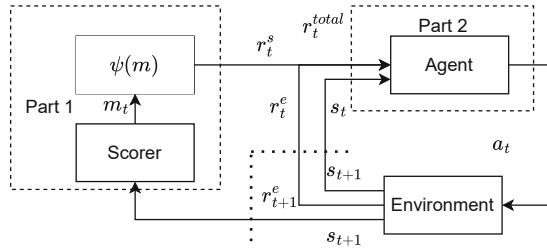


Fig. 2. RLFC introduces an extra part to the RL framework, In part 1, a scorer has previous $k - 1$ historical states and computes a score m according to $k - 1$ historical states and 1 future state, then m is sent to a scorer function ψ , ψ maps a score m to a reward from scorer, in part 2, other than s_t , r_t , the agent gets an extra reward r_t^s from the scorer function

4.1 Pre-train the Scorer

In this section, we discuss how to pre-train a scorer ϕ from demonstrations. A good scorer leads to a stabler and faster training process. We first show how to pre-train this scorer.

Some works send feedback to an agent by an expert pressing the keyboard [8], which costs much human work. In this work, the scorer needs a few expert demonstration clips, we extract a few expert data clips from D4RL, and then train the scorer ϕ . The training details are described in Algorithm 1.

In Algorithm 1, a scorer ϕ is parameterized by θ , unlike we discussed in proof A, this algorithm uses 0 and 1 as update targets. The intuitive behind this is that parameterized scorer ϕ_θ may make errors, and if the training process for this scorer is exactly like the process in proof A, the scorer will only work if we calculate all by math or the expert demonstrations is enough.

Consider that this scorer is trained with two targets 0 and 1, this scorer will give a score only when it encounters a familiar scene. Any strange scene will get a 0 and cannot disturb the training process. This means we don't need to collect much data for the training process but a few key demonstrations, which can enable us to train.

4.2 Train the Agent with the Scorer

When got the scorer, we can train any RL agent with this scorer, details are in Algorithm 2.

Algorithm 1. Training Scorer

Require: clip length k ; a set \mathcal{P}^e with n expert demonstration clips with length k ; a set with all states \mathcal{S}_{all} .

```

Initialize Scorer Network  $\phi$  and its parameter  $\theta$ ;
for  $idx_{out} = 0$ ;  $idx_{out} < |\mathcal{P}^e|$ ;  $idx_{out} \leftarrow idx_{out} + 1$  do
     $\mathcal{H} \leftarrow \mathcal{P}^e[idx_{out}]$ 
     $\mathcal{H}_{tmp} \leftarrow []$ 
    for  $idx_{in} = 0$ ;  $idx_{in} < |\mathcal{H}|$ ;  $idx_{in} \leftarrow idx_{in} + 1$  do
         $\mathcal{H}_{tmp} \leftarrow \mathcal{H}_{tmp} + [\mathcal{H}[idx_{in}]]$ ;
        if  $len(\mathcal{H} < k)$  then
            Continue;
        end if
         $m \leftarrow \phi_\theta(\mathcal{H}_{tmp}[-k :])$ ;
         $\mathcal{L}_p = LossFn(m, 1)$ ;
         $s_{rand} \leftarrow$  sample from  $\mathcal{S}_{all}$ ;
         $m \leftarrow \phi_\theta(\mathcal{H}_{tmp}[-k : -1] + [s_{rand}])$ ;
         $\mathcal{L}_n = LossFn(m, 0)$ ;
         $\mathcal{L} \leftarrow \mathcal{L}_p + \mathcal{L}_n$ ;
    end for
    UpdateScorer  $\theta \leftarrow \theta - \alpha_\phi \Delta_\phi \mathcal{L}$ ;
end for

```

In Algorithm 2 and 3, we use Soft Actor-Critic(SAC) [5] as an example to show how this scorer works. As in SAC, we collect K experiences at first, including old states, new states, actions, rewards, and dones, at this collecting step, all gradients are closed, in addition to those, RLFC will compute scores for each state list and put those scores into replay buffer. In the training step, on the one hand, like the vanilla SAC, RLFC uses old states to compute actor loss using Eq. 2:

$$T^\pi Q(s_t, a_t) \triangleq r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)} [V(s_{t+1})] \quad (2)$$

where

$$V(s_t) = \mathbb{E}_{a_t \sim \pi} [Q(s_t, a_t) - \log \pi(a_t | s_t)] \quad (3)$$

On the other hand, the critic update in RLFC is different from that in SAC. RLFC adds scores to critic target Q values, changes it from Eq. 4 to Eq. 5,

$$target_q_values_{sac} = r_{env} + (1 - dones) * next_q_values \quad (4)$$

$$target_q_values_{rlfc} = r_{scorer} + r_{env} + (1 - dones) * next_q_values \quad (5)$$

and that explains why we don't add scores to the actor updating process. Because of the score, the target of the critic network now becomes Eq. 7

$$V(s_t) \triangleq r_{scorer_0} + r_{env_0} + \gamma(r_{scorer_1} + r_{env_1}) + \dots + \gamma^n(r_{scorer_t} + r_{env_t}) \quad (6)$$

$$V(s_t) \triangleq \sum_{t=0}^n (\gamma^t(r_{scorer_t} + r_{env_t})) \quad (7)$$

Algorithm 2. Training Agent Using Scorer(on SAC example)

Require: A well-trained scorer ϕ ; a mapping function ψ to map score to reward from scorer; training epochs number N_{epoch} ; T round per epoch; batch size B , experience number K for each collection; length of historical state len .

```

Initialize Actor and Critic Network  $\pi$  and  $v$  and their parameters  $\theta_\pi$  and  $\theta_v$ ;
 $\mathcal{D} \leftarrow []$ ;
for  $idx_{epoch} = 0$ ;  $idx_{epoch} < N_{epoch}$ ;  $idx_{epoch} \leftarrow idx_{epoch} + 1$  do
    // collect K experiences, see algorithms 3
     $\mathcal{D} \leftarrow \mathcal{D} + CollectRollouts(\phi, \psi, K, len)$ ;
    for  $idx_{train} = 0$ ;  $idx_{train} < T$ ;  $idx_{train} \leftarrow idx_{train} + 1$  do
         $s_{old}, s_{new}, actions, r_{env}, r_{scorer}, done \leftarrow$  sample from  $\mathcal{D}$ ;
        // We don't change the actor's updating process, the reason is provided later
         $\theta_\pi \leftarrow ActorUpdate(s_{old}, actions)$ ;
         $\theta_v \leftarrow CriticUpdate(s_{old}, s_{new}, r_{scorer} + r_{env}, done, actions)$ 
    end for
end for

```

Algorithm 3. CollectRollouts

Require: A well-trained scorer ϕ ; a mapping function ψ to map score to reward from scorer; experience number K for each collection; length of historical state len .

```

 $\mathcal{D} \leftarrow []$ 
state  $\leftarrow env.reset()$ ;
if  $\mathcal{H}_{len-1} == []$  then
     $\mathcal{H}_{len-1} \leftarrow [state, state, state]$ ; // at the very beginning,  $\mathcal{H}_{len-1}$  is empty
end if
for  $idx = 0$ ;  $idx < K$ ;  $idx \leftarrow idx + 1$  do
    action  $\leftarrow \pi_\theta(state)$ ;
    statenew, rewardenv, done, info  $\leftarrow env.step(action)$ ;
    rewardscorer  $\leftarrow \psi(\phi(\mathcal{H}_{len-1} + [state_{new}]))$ ;
     $\mathcal{D} \leftarrow \mathcal{D} + [[state, state_{new}, action, reward_{env}, reward_{scorer}, done]]$ ;
    state  $\leftarrow state_{new}$ ;
end for
return  $\mathcal{D}$ 

```

causing the actor to become Eq. 8.

$$\mathcal{T}^\pi Q(s_t, a_t) \triangleq r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)}[V(s_{t+1})] \quad (8)$$

where

$$V(s_t) = \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \log \pi(a_t | s_t)] + r_{scorer_t} \quad (9)$$

if we add a score to the actor's update process, the actor will get a double score, which is not what we want. RLFC value function is trained to minimize the squared residual error

$$\mathcal{J}(\theta_v) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[\frac{1}{2} (V_{\theta_v}(s_t) - (\mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \log \pi(a_t | s_t)] + r_{scorer_t})) \right] \quad (10)$$

The gradient of Eq. 10 can be estimated with

$$\hat{\nabla}_{\theta_v} \mathcal{J}_v(\theta_v) = \nabla_{\theta_v} V_{\theta_v}(s_t)(V_{\theta_v}(s_t) - Q(s_t, a_t) + \log \pi(a_t | s_t) - r_{scorer_t}) \quad (11)$$

which means that this reward from the scorer will affect the gradients and thus affect the learning process.

5 Experiments

In this section, we show that RLFC gives a more stable training process and faster learning speed in the early stage than popular frameworks in various Gym environments, including continuous control tasks and discrete control tasks. The experimental code can be found at <https://github.com/ZhuShaoQiang/RLFC> [19].

The goal of our experiment is to understand how RLFC works, and to see the agent's performance under this RLFC framework. We will experiment on the Cliff Walking environment at first and then dive into more complicated environments such as MuJoCo. CliffWalking is a Toy environment used as a example to show the performance of this RLFC. MuJoCo, Multi-Joint dynamics with Contact, is a physics engine, can be used to train an agent who can control those robots, MuJoCo includes environments such as Ant, the agent needs to control this ant to run.

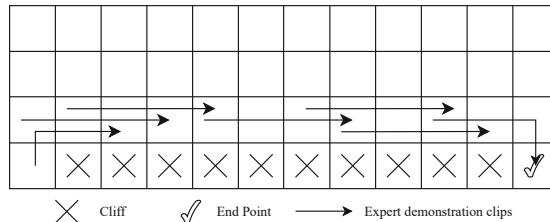


Fig. 3. As shown in the picture, we made some expert demonstration clips to train the scorer

5.1 Discrete Action Space

The most well-known algorithm in discrete control tasks is DQN [17], we first show RLFC's performance in discrete environments. In this setting, we use the Cliff Walking environment, which is a classic control task, the goal of it is to control the agent to go to the final point as soon as possible.

Train the Scorer. Considering this environment is simple and small, we manually set some expert demonstration clips to train the scorer, as shown in Fig. 3, the green lines are expert demonstration clips.

Train the Agent. Then we train the agent under a well-trained scorer using Algorithms 2 in the DQN version, and the result is shown in Fig. 4.

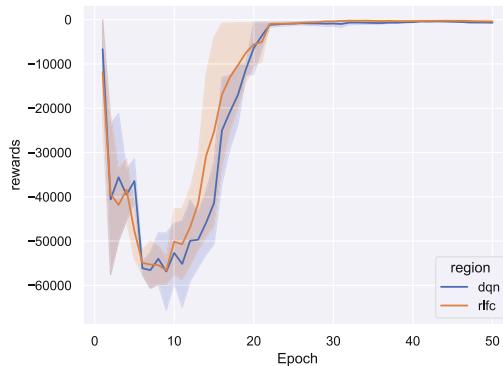


Fig. 4. As shown in the picture, RLFC agent gets faster than DQN

5.2 Continous Action Space

We train a scorer and an agent under the Mujoco Ant-v4 environment.

Train the Scorer. We use the existing dataset D4RL [4] to train this scorer using Algorithm 1, but we only use one percent of the whole expert dataset to simulate the small number of expert demonstration clips. The time length k of historical states is 4 here.

Train the Agent Under Scorer. Then we train the agent under a well-trained scorer using Algorithm 2. Figure 5 compares SAC and RLFC under 3 runs. We can see in the picture, that the RLFC framework not only learns faster but also more stable than SAC algorithms. All two results show that this RLFC framework can boost learning speed and enhance stability during training.

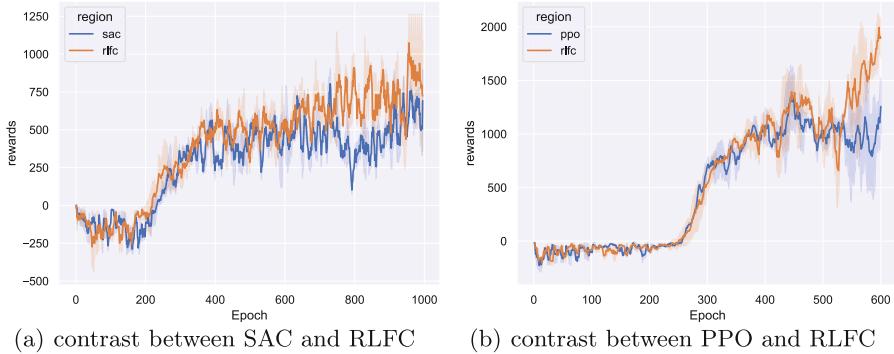


Fig. 5. We show the comparison of the SAC algorithm, the PPO algorithm, and RLFC(ours), from the picture above, we can see that our framework not only gets an early rise but is stable at every stage, and the RLFC framework has no dramatic performance drop during training

6 Conclusion and Discussion

We present RLFC, a plug-in algorithm for every sort of RL algorithm framework that provides stability and training speed while the agent interacts with the environment, RLFC combines the advantages of online and offline RL learning, it can be trained on more unseen states, which will enhance its ability of generalization, and it is guided by the offline data. Also, RLFC alleviates the disadvantages of online and offline RL learning, which are sparse reward problem, unstable learning process, and poor ability of generalization. Our experiment result shows that RLFC framework can work at both continuous and discrete control tasks.

According to the training process illustrated in all algorithms, if the domain knowledge is not enough, this scorer will give 0 score at almost all step, this won't influence or will boost slightly the original algorithm's performance.

Acknowledgement. PI would like to thank my teachers Professor Haiwei Pan and Vice Professor Kejia Zhang for their kindness and guidance and thank stable-baselines [7] for saving my time in using SAC, DQN, and PPO algorithms.

Disclosure of Interests. The work was supported by the National Natural Science Foundation of China under (Grant No. 62072135) and the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

A Proof

Proof of Theorem 1. Assume any MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma)$ with $u (u \geq 1, u \in \mathbb{Z})$ expert solutions, and the i -th expert solution's state sequence is $\mathcal{H}^{e_i} = <$

$s_1^{e_i}, s_2^{e_i}, \dots, s_n^{e_i} >$, ($1 \leq i \leq u, i \in \mathbb{Z}$), the set of state sequences of all expert solutions is $\mathcal{P}^e = \{\mathcal{H}^{e_1}, \mathcal{H}^{e_2}, \dots, \mathcal{H}^{e_u}\}$.

For a state sequence $\mathcal{H}^{\pi_j} = < s_1^{\pi_j}, s_2^{\pi_j}, \dots, s_n^{\pi_j} >$ generated by any policy π_j in policy space Π ($1 \leq j \leq |\Pi|$), the set of its subsequence with length k is $\mathcal{P}_{subk}^{\pi_j} = \{\mathcal{H}_{k_1}^{\pi_j}, \mathcal{H}_{k_2}^{\pi_j}, \dots, \mathcal{H}_{k_{n-k+1}}^{\pi_j}\}$, $n \geq k \geq 2$. The set of $\mathcal{P}_{subk}^{\pi_j}$ generated by all policies in Π is $\mathcal{P}_{subk}^{\pi} = \bigcup_{j=1}^{|\Pi|} \mathcal{P}_{subk}^{\pi_j}$.

Given two sequences with length k and n , it is well-known that the time complexity of determining the length of the longest common subsequence is $O(nk)$, assume function \mathcal{F} can solve this problem with time complexity $O(nk)$, the input of \mathcal{F} is a state sequence \mathcal{H}_k with length k , \mathcal{F} compute the longest matching length between \mathcal{H}_k and all the expert sequence \mathcal{H}^{e_u} in \mathcal{P}^e , the output of \mathcal{F} is the longest length l ($l \in \mathbb{Z}^+, 0 \leq l \leq k$), the complexity of \mathcal{F} is $O(|\mathcal{P}^e|*nk)$.

Construct a scorer ϕ , for any input \mathcal{H}_k with length k :

$$\phi(\mathcal{H}_k) = \frac{\mathcal{F}(\mathcal{H}_k)}{k}. \quad (12)$$

In this case, ϕ is a satisfactory scorer that meets requirements. \square

References

1. Bojarski, M., et al.: End to end learning for self-driving cars. arXiv preprint [arXiv:1604.07316](https://arxiv.org/abs/1604.07316) (2016)
2. Buffet, O., Pietquin, O., Weng, P.: Reinforcement learning (2020)
3. Degris, T., Pilarski, P.M., Sutton, R.S.: Model-free reinforcement learning with continuous action in practice. In: 2012 American Control Conference (ACC), pp. 2177–2182. IEEE (2012)
4. Fu, J., Kumar, A., Nachum, O., Tucker, G., Levine, S.: D4rl: datasets for deep data-driven reinforcement learning (2021)
5. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor (2018)
6. Hasselt, H.: Double q-learning. Adv. Neural Inf. Process. Syst. **23** (2010)
7. Hill, A., et al.: Stable baselines (2018). <https://github.com/hill-a/stable-baselines>
8. Li, G., He, B., Gomez, R., Nakamura, K.: Interactive reinforcement learning from demonstration and human evaluative feedback. In: 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 1156–1162. IEEE (2018)
9. Mnih, V., et al.: Playing atari with deep reinforcement learning (2013)
10. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
11. Ng, A.Y., Russell, S., et al.: Algorithms for inverse reinforcement learning. In: ICML (2000)
12. Peng, X.B., Kanazawa, A., Malik, J., Abbeel, P., Levine, S.: SFV: reinforcement learning of physical skills from videos. ACM Trans. Graph. (TOG) **37**(6), 1–14 (2018)
13. Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P.: Trust region policy optimization (2017)

14. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
15. Silver, D.N., et al.: Mastering the game of go with deep neural networks and tree search. *Nature* **529**(7587), 484–489 (2016)
16. Torabi, F., Warnell, G., Stone, P.: Behavioral cloning from observation. arXiv preprint [arXiv:1805.01954](https://arxiv.org/abs/1805.01954) (2018)
17. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
18. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1995–2003. PMLR (2016)
19. Zhu, S.: RLFC (2024). <https://github.com/ZhuShaoQiang/RLFC>



Self Supervised Contrastive Learning Combining Equivariance and Invariance

Longze Yang, Yan Yang^(✉), and Hu Jin^(✉)

School of Computer Science and Technology, Heilongjiang University,
Harbin 150080, China
 {2221945,jinhu}@s.hlju.edu.cn, yangyan@hlju.edu.cn

Abstract. Current self-supervised representation learning methods are mainly based on contrastive learning and proxy tasks. These methods acquire semantically rich features by contrasting samples with invariant transformations (positive pairs) against other samples (negative pairs), and simply discard transformations that degrade performance when used as invariances. However, using only invariant transformations often leads to an over-reliance on invariant transformations, which affects the generalisation ability and robustness of the model, while the large number of negative sample pairs in contrast learning imposes a huge computational overhead. In order to address these issues, we reduce the dependence on invariant transformations by transforming the discarded invariant transformations into equivariant transformations. In contrast learning, we reduce the computational overhead by using only positive pairs to obtain semantically rich features. Specifically, we enhance feature semantic quality by encouraging certain transformations to exhibit non-trivial equivariance on samples of invariant transformations in the form of a proxy task, while preserving original transformation invariance. The model learns the invariant transformations further by learning equivariance at the same time, and our approach can improve the accuracy of the model without changing the structure of the original model. Experimental results show that significant improvements are obtained on several benchmark datasets.

Keywords: transformations · invariance · equivariance

1 Introduction

In today’s deep learning landscape, a major challenge stems from the reliance on large annotated training datasets, prompting the surge in interest in self-supervised learning (SSL). SSL leverages unlabeled datasets to extract information without the need for label information. Instead of using labeled data, SSL employs some form of context or pretext task to predict certain aspects of the data. This approach yields experimental performance comparable to, or even surpassing, supervised learning in various computer vision tasks, garnering widespread attention.

SimCLR [6], BYOL [17], and MOCO [20] have become milestone models in self-supervised learning by utilizing a Siamese structure and invariant transformations. Their influence has been so significant that subsequent models have been, to varying degrees, impacted by their approaches. The Siamese structure involves obtaining multiple augmented views from a single image as input and feeding them into different networks, minimizing their distance in the embedding space. [8] Yet, the most advanced SSL methods today predominantly encourage invariant-enhancing transformations on input information. The most common method is contrastive learning, which learns invariant features through a binary classification task. Specifically, given a set of images, the same image undergoes different transformations like color jittering, random grayscale, Gaussian blur, and solarization to construct positive pairs, while other data points serve as negative pairs [6, 20]. However, this loss function is highly sensitive to the quantity and quality of negative samples, often leading to the generation of numerous false negative samples when other images are set as negative pairs. This can adversely affect accuracy, prompting the emergence of models that solely utilize positive pairs.

Previous studies have demonstrated the importance of transformations and evaluated various transformation methods [27]. Among them, rotation, despite preserving semantic information, is detrimental to contrastive learning. This led to the adoption of invariant transformations for data augmentation in SSL [5, 30], with detrimental transformations being simply removed during usage. However, this does not imply that four-fold rotations are harmful to SSL. Rotation can be considered as an effective proxy task for evaluating the quality of feature representations generated by contrastive learning [12]. Research indicates that training neural networks to predict image rotations, rather than simply making them invariant to rotations, can yield excellent image representations, even across multiple rotation angles [16]. These findings suggest that the choice between sensitivity and insensitivity to specific transformations in feature design may significantly impact the performance of downstream tasks [15].

The equivariance and invariance are also known as sensitivity and insensitivity, respectively. Invariance is a trivial instance of equivariance and is often considered a key property of features. equivariance is concerned with how the image transforms when it is transformed, and it can be learned empirically from the data. Invariance is concerned with being unaffected by the image transformations so as to get the same results as the original image. Both of these concepts will be explained in more detail in Sect. 3.1 [25]. We believe that good features should incorporate both invariance and equivariance.

Therefore, we decided to keep the invariant transforms with good performance and change the invariant transforms that are detrimental to accuracy (e.g. rotations) to equivariant transforms. The structure of Siamese uses only positive samples for invariant self-supervised learning, and introduces a simple equivariant transform as an additional agent task on the samples using the original invariant transform. This task performs invariant self-supervised learning by predicting the transformation of the input. In our approach, equivariant

and invariant learning interact with each other, complementing each other by accumulating feature extraction capabilities for equivariant learning in invariant learning, and further learning about the effects of invariant transformations on pictures in equiariant learning. Our method does not need to modify the original invariant self-supervised learning model, so the generalisation performance is very good, bringing better experimental performance.

In summary, our contributions are the following:

- We propose the E-BYOL(Equivariant BYOL), which integrates equivariance and invariance, leveraging the complementary advantages of both to achieve a significant performance improvement.
- Our method is plug-and-play, we abandon the simplistic approach of directly imparting equivariance to the model and instead choose to integrate invariance into the learning of equivariance, thus learning equivariant transformations and achieving a mutually beneficial relationship between equivariance and invariance.
- We show experimentally that our method can lead to a significant improvement in the accuracy of the model and speed up the convergence of the model, and we experimentally analyse the effect of our method on invariance.

2 Background and Related Work

2.1 Contrastive Learning

Contrastive learning is a widely used approach in machine learning, where the core idea involves learning and inference through comparing the similarity and dissimilarity between different samples, aiming to pull together positive pairs and push apart negative pairs in a vector space. This method finds extensive applications in self-supervised learning, unsupervised learning, and supervised learning [3]. The aim is to learn the data representation by maximising the similarity between related samples and minimising the similarity between unrelated samples. Typically, a high-degree, custom rule is employed to generate positive and negative samples, among which the Siamese structure in contrastive learning has demonstrated excellent performance in downstream tasks, sometimes even surpassing certain supervised models [6, 7].

In the evolution of contrastive learning, the emergence of SimCLR [6] has laid a robust foundation for contrastive visual representation learning. SimCLR utilizes more data augmentation techniques and adds a prediction head to learn non-linear transformation layers, significantly enhancing the quality of learned representations with non-linear transformation heads [6]. Subsequently, Moco [20], by employing a queue to store negative samples, forms a large and consistent dictionary to aid contrastive learning. It also proposes a momentum encoder, a method still used in the latest contrastive learning approaches.

In further developments, the emergence of BYOL [17], which learns without negative samples, has sparked intense debate, as negative samples play a crucial role in contrastive learning. If only positive samples are considered, the objective

function will be greatly limited, and the learning goal of the model will become singular. Essentially, the goal is to ensure that similar objects are as close as possible in the feature space. In such a scenario, the model may opt for a simplistic way to achieve this goal, even mapping all inputs to the same output. Consequently, all feature representations will become identical, and the contrastive learning loss will converge to zero. However, by introducing negative samples, the model gains more learning constraints. The presence of negative samples compels the model to differentiate between different objects, as mapping all samples to the same feature representation will lead to an infinitely large loss function for negative samples, which is unacceptable [14]. Therefore, the introduction of negative samples can prevent the model from falling into the so-called “model collapse”, where it learns a simplified but useless feature representation, a phenomenon commonly referred to as model collapse in literature [14]. The authors of BYOL [17] argue that the additional projector and momentum encoder in online networks are crucial for avoiding the collapse without negative samples [14].

2.2 Pretext Task

The rise of self-supervised learning is attributed to the high cost of training deep learning models with a large number of manually labeled samples. The introduction of self-supervised learning aims to break the limitation of manually labeled samples, allowing learning even in the absence of label information, and extracting useful feature representations. The core challenge lies in the design of pretext tasks, which can be understood as indirect tasks designed to achieve specific training objectives. When dealing with static images, one of the most widely used pretext tasks is instance discrimination. This involves extracting different views of images using data augmentation techniques such as image cropping, rotation, color jittering, Sobel filtering, etc. [29]. The extracted views are treated as positive samples, while other different samples are treated as negative samples, thus learning more discriminative feature representations. Among these, Siamese networks are widely used for instance discrimination. Pseudo-labeling is another type of pretext task, where common methods for generating pseudo-labels include image rotation, colorization, completion, relation prediction, etc. [2].

Based on previous studies, we found that using rotation for image augmentation and using image rotation as a pretext task both resulted in unsatisfactory experimental performance. We believe that relying solely on one method is too narrow-minded; it is important to encourage both equivariance and invariance in augmentation [12]. While most previous models only include instance discrimination as a pretext task, we combine image rotation with instance discrimination. What sets our approach apart is that it not only encourages the model to be insensitive to certain transformations (invariance) but also encourages sensitivity to other transformations (equivariance). Moreover, our approach does not require the model to simultaneously exhibit insensitivity and sensitivity to specific transformations. This means that our method can more flexibly adapt to

different types of transformations and provide more comprehensive representation learning in various scenarios.

3 Method

3.1 Preliminaries

The concepts of invariance and equivariance are both defined within the context of data transformations, often referred to as insensitivity and sensitivity, respectively [1, 22, 25]. Let G be a group of transformations, where $g \in G$, and let $T_g(x)$ denote the function used to transform an input image x . Furthermore, let f denote the encoder network that computes the feature representation, the feature representation of image x is denoted as $f(x)$.

Invariance is a trivial instance of equivariance [32], invariance encourages the output $f(x)$ not to vary with T_g , invariance is defined as follows

$$f(T_g(x)) = f(x) \quad (1)$$

self-supervised learning of invariance encourages the model to recognize transformed samples, aiming for the transformed samples to produce the same representation in the encoder as the untransformed samples [26].

Equivariance is a generalisation of invariance [32], the definition of equivariance is as follows

$$f(T_g(x)) = T'_g f(x) \quad (2)$$

where, $T'_g f(x)$ represents the result of the output $f(x)$ after undergoing a fixed transformation. Equivariance requires the model's output to change correspondingly with transformations applied to the input data, meaning they can predict the transformations applied to the samples, where the manner of transformation is determined by a fixed transformation $T_g(x)$ [4, 10, 25].

In invariance, T'_g exists as a identity function, i.e., $T'_g f(x) = f(x)$, and we say that f is insensitive to g and g is an invariant transformation of f . In equivariance, T'_g does not exist as a identity function, then f is sensitive to g and g is an equivariant transformation of f [25, 32].

3.2 Model

Our model is built upon the basic framework of BYOL [17] and improves upon it by incorporating four-fold rotation as an additional proxy task, as Fig. 1 illustrates. Since BYOL [17] only uses positive samples, it results in fewer Comparative learning samples compared to typical self-supervised learning approaches [14], the additional samples generated by four-fold rotation help offset the limitation of fewer training samples. Moreover, the computational overhead introduced by incorporating four-fold rotation is manageable, given that BYOL [17] already incurs lower computational costs compared to conventional self-supervised learning methods.

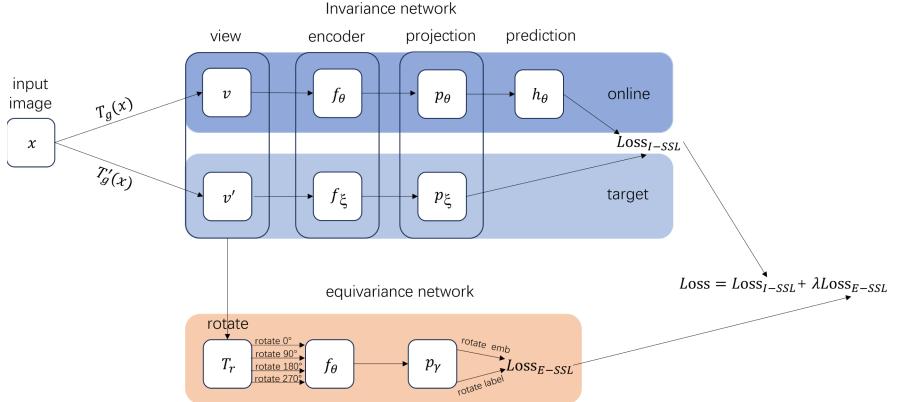


Fig. 1. Our model consists of an invariance network and an equivariance network. The invariance network comprises an online network and a target network. The equivariance network utilizes views v and v' transformed by the invariance network as samples.

Our method combines both invariance and equivariance representations to better learn semantically rich features. For invariance, we use the traditional comparative learning method BYOL [17], utilizing a momentum encoder with momentum-updated parameters. While maintaining invariance to other transformations, we introduce equivariance through a proxy task. The equivariant network shares the encoder of the invariant network and is tasked with predicting the transformations of features using its own prediction head, denoted as p_γ , which encourages equivariant prediction. Notably, our method does not require modifications to the invariant model, facilitating better transferability to other models.

The model employs two neural networks for learning: the invariance network and the equivariance network. The invariance network is trained using two neural networks: the online network and the target network. The online network, with parameters denoted as θ , comprises an encoder f_θ , a projection head p_θ , and a prediction head h_θ . The target network, with parameters denoted as ξ , consists of an encoder f_ξ and a projection head p_ξ . The parameters ξ of the target network are updated using exponential moving averages of the parameters θ of the online network [20]. The update formula is as follows:

$$\xi = \tau \xi + (1 - \tau) \theta \quad (3)$$

For the aspect of invariance, given a set of images D , uniformly sample an image $x_i \in D$, let G be a set of transformations, where $g \in G$. Let $T_g(x_i)$ denote the function used to transform input image x_i , generating two augmented views from image x , one applying the image transformation T_g and the other applying the image transformation T_g' , $v_i = T_g(x_i)$, $v'_i = T_g'(x_i)$. We feed v_i and v'_i into the invariance network. We then pass v_i and v'_i through the f_θ and f_ξ encoders respectively, $y_{i\theta} = f_\theta(v_i)$, $y_{i\xi} = f_\xi(v'_i)$. We pass $y_{i\theta}$ and $y_{i\xi}$ through

the g_ξ and g_θ projection heads respectively, $z_{i\theta} = p_\theta(y_{i\theta})$, $z_{i\xi} = p_\xi(y_{i\xi})$. Then, the output $z_{i\theta}$ is passed through the h_θ head to obtain $h_\theta(z_{i\theta})$, normalization is performed on both $h_\theta(z_{i\theta})$ and $z_{i\xi}, \bar{h}_\theta(z_{i\theta}) = h_\theta(z_{i\theta})/\|h_\theta(z_{i\theta})\|_2$ and $\bar{z}'_{i\xi} = z'_{i\xi}/\|z'_{i\xi}\|_2$, and the loss is calculated:

$$l_i = \mathcal{L}_{\theta,\xi} = \|\bar{h}_\theta(z_{i\theta}) - \bar{z}'_{i\xi}\|_2^2 = 2 - 2 \cdot \frac{\langle h_\theta(z_{i\theta}), z'_{i\xi} \rangle}{\|h_\theta(z_{i\theta})\|_2 \cdot \|z'_{i\xi}\|_2}. \quad (4)$$

$$Loss_{I-SSL} = -\frac{1}{N} \sum_{i=1}^N l_i \quad (5)$$

The equivariance network consists of the encoder f_θ of the invariance network and the projection head p_γ . It is important to note that the projection head p_γ of the equivariance network is different from the projection head of the invariance network. Specifically, the projection head of the equivariance network is equivalent to being composed of two projection heads from the invariance network.

Input both v_i and v'_i into the equivariant network, let $T_r(x_i)$ denote the function used to rotate, rotate v_i and v'_i by 90, 180, and 270, respectively obtain $v_{in} = T_r \frac{n\pi}{2}(v_{in}), v'_{in} = T_r \frac{n\pi}{2}(v'_{in}), \{n = 0, 1, 2, 3\}$. Pass v_{in} and v'_{in} into the f_θ encoder separately, $z_{in} = f_\theta(v_{in}), z'_{in} = f_\theta(v'_{in})$, pass z_{in} and z'_{in} separately into the projection head p_γ to output the predictions p_{in} for z_{in} and p'_{in} for z'_{in} . Finally, compute the equivariance loss. The formula is as follows:

$$Loss_{E-SSL} = -\frac{1}{N} \sum_i \sum_{n=0}^3 (y_{ic} \log(p_{in}) + y'_{in} \log(p'_{in})) \quad (6)$$

where i represents the number of samples, y_{in} is the label of the i -th sample category n , and p_{in} and p'_{in} are the probabilities of the i -th different transformation sample category n .

We define the parameter λ as the strength of the equivariance loss, which is calculated as follows:

$$Loss = Loss_{I-SSL} + \lambda Loss_{E-SSL} \quad (7)$$

The integration of invariance and equivariance in the model manifests as separate networks, yet they are interconnected. Unlike other models that simply combine them, in our method, the samples used in the equivariance network are not simply cropped from the original samples, but are obtained as transformations from the invariance network. This allows the encoder to learn invariance iteratively during equivariance learning, acquiring additional features to predict equivariant transformations. This complementary relationship between equivariance and invariance provides the encoder with more direct optimization directions.

Algorithm 1. PyTorch-style pseudocode for combining equivariance and invariance.

```

#  $f_\theta$ : online network encoder
#  $f_\xi$ : target network encoder
#  $p_\theta$ : projector online network for I-SSL
#  $p_\xi$ : projector target network for I-SSL
#  $h_\theta$ : predictor online network for I-SSL
#  $p_\gamma$ : projector network for E-SSL
#  $ssl\_loss$ : loss function for I-SSL
#  $\lambda$ : weight of the E-SSL

for  $x$  in loader :
    # Obtain the transformed view
     $v, v' = \text{augment}(x)$ 

    # loss
    loss_invariance =  $ssl\_loss(h_\theta(p_\theta(f_\theta(v))), (p_\xi(f_\xi(v'))))$ 
    # Construct rotated views and labels
    rotated_images, rotated_labels =  $\text{rotate\_images}(v, v')$ 
    logits =  $p_\gamma(f_\theta(\text{rotate\_images}))$ 
    # rotation prediction
    loss_equivariance =  $\text{CrossEntropyLoss}(\text{logits}, \text{rotated\_labels})$ 
    loss = loss_invariance +  $\lambda * loss\_equivariance$ 

    # optimization step
    loss.backward()
    optimizer.step()
end for

```

4 Experiments

4.1 Experimental Setups

Datasets. For the downstream task we use four datasets to evaluate our method, namely the CIFAR-10 and CIFAR-100 datasets [23], the STL-10 dataset [9], and the Tiny ImageNet dataset [24].

Setting. The goal of our experiments is to compare BYOL improved by our method with BYOL alone [17] and BYOL with enhanced contrast views [28] as well as with BYOL using a module that uses meta-integrated regularisation [19]. We used the same encoder architecture, ResNet-18, for all comparison methods. $L2$ normalization was applied to the latent space features in each method. We initialized the exponential moving average with a starting value of 0.99.

Implementation Details. In terms of implementation, we employed the Adam optimizer for datasets of moderate size [11]. Across all compared methods, we

maintained consistency in the number of epochs and the learning rate schedule. Specifically, for CIFAR-10 and CIFAR-100, we conducted training for 1000 epochs with a learning rate of 3×10^{-3} ; for Tiny ImageNet, we conducted training for 1000 epochs with a learning rate of 2×10^{-3} ; and for STL-10, we extended the training to 2000 epochs with a learning rate of 2×10^{-3} . We applied learning rate warm-up for the initial 500 iterations of the optimizer and scheduled a 0:2 learning rate decay in the last 50 and 25 iterations. The projection head’s hidden layer dimensionality, denoted as $g(\cdot)$, was set to 1024 [13]. As for the prediction head of the equivariant network, it consisted of a two-layer MLP, with the final layer serving as a linear classifier, featuring a hidden dimension of 2048. After each linear layer, we incorporated layer normalization [21] followed by ReLU activation.

Image Transformation Details. The random crop size we extract ranges from 0.2 to 1.0 times the original area, while the random aspect ratio varies between 3:4 and 4:3 of the original aspect ratio, a commonly used data augmentation technique [6]. Additionally, we apply horizontal flipping with a probability of 0.5. Finally, we utilize a configuration of (0.4, 0.4, 0.4, 0.1), with color jittering probability set to 0.8 and grayscale probability set to 0.1. For ImageNet and ImageNet-100, we adhere to the details outlined by BYOL [17]: crop size ranging from 0.08 to 1.0, stronger color jittering (0.8, 0.8, 0.8, 0.2), grayscale probability set to 0.2, and Gaussian blur probability set to 0.5.

Evaluation Protocol. The prevailing assessment protocol for unsupervised feature learning entails preserving the network encoder post unsupervised pretraining, followed by the training of a supervised linear classifier atop it. Specifically, this classifier comprises a fully connected layer followed by softmax, integrated onto $f(\cdot)$ subsequent to the removal of the projection head $p(\cdot)$. Across all experiments, we trained the linear classifier with labeled training sets of each specific dataset for 500 epochs, employing the Adam optimizer, sans data augmentation. The learning rate decayed exponentially from 10^{-2} to 10^{-6} , weight decay was set at 5×10^{-6} [13]. Furthermore, in our experiments, we incorporated the accuracy of the k-nearest neighbor classifier (k-nn, $k = 5$) [18]. This classifier’s advantage lies in its absence of requisite additional parameters or training, alongside its deterministic nature.

4.2 Downstream Tasks

In our experiments, conducted on the CIFAR-10, CIFAR-100, STL-10, and Tiny ImageNet datasets, we employed different methods: BYOL [17], Use C-Crop to get a better contrast view of BYOL($C-Crop$) [28], BYOL($CompMod$) [19] using meta-synthesis regularisation module, and our proposed model(E-BYOL). The experimental results are presented in Table 1. Our model demonstrates superior performance in terms of both linear classification and KNN accuracy across these four datasets, affirming the effectiveness of our approach in enhancing BYOL

[17]. The integration of equivariance and invariance leads to improved model performance, while the higher KNN accuracy suggests better prediction of class labels for unseen data samples, along with effective capture of semantic similarity and structural information inherent in the data, thereby acquiring rich semantic representations.

Table 1. Classification accuracy (top 1) of a linear classifier and a 5-nearest neighbors classifier for different loss functions and datasets with a ResNet-18 encoder

Method	CIFAR-10		CIFAR-100		STL-10		Tiny ImageNet	
	linear	5-nn	linear	5-nn	linear	5-nn	linear	5-nn
BYOL [17]	91.73	89.45	66.60	56.82	91.99	88.64	51.00	36.24
BYOL($C - Crop$) [28]	92.54	90.76	64.62	54.33	92.42	89.98	47.23	31.72
BYOL(CompMod) [19]	93.85	91.53	68.74	58.01	94.73	89.88	53.51	37.95
E-BYOL	93.97	92.04	68.83	58.47	94.45	89.86	53.62	38.16

4.3 Convergence Rate

Table 2. The convergence speed and classification accuracy of linear classifier and 5-nearest neighbor classifier using ResNet-18 encoder for different model and epochs. Top 1 and 5 correspond to the accuracy of a linear classifier

Method	300 epochs			700 epochs			1000 epochs		
	linear	top-5	5-nn	linear	top-5	5-nn	linear	top-5	5-nn
BYOL [17]	89.47	99.71	86.68	88.64	99.70	89.11	91.74	99.82	89.45
E_s -BYOL [17]	92.62	99.86	90.40	93.10	99.85	91.30	93.23	99.89	91.61
E-BYOL	92.72	99.81	90.87	93.97	99.91	91.94	93.89	99.89	92.01

On the CIFAR-10 dataset, we compared the performance of our model E-BYOL, with BYOL and E_s -BYOL (a simple fusion of equivariance and invariance) across different epochs. Specifically, E_s -BYOL is directly cropping the original samples (without transformations) and feeding them into the equivariant network to obtain classification accuracy. As shown in Table 2, E-BYOL exhibited significant advantages. Our method achieved rapid convergence of the model at around 700 epochs, while E_s -BYOL, with its simple mixture strategy, and BYOL required a longer time, approximately 1000 epochs, to converge. Furthermore, BYOL demonstrated an even slower convergence rate compared to E_s -BYOL, requiring even more epochs to reach convergence. Therefore, our E-BYOL method not only outperforms the other two methods in convergence

speed but also demonstrates its efficiency and superiority. From Table 2, it can be seen that the impact of using invariant transformation samples in equivariant networks on convergence is crucial, and equivariant networks improve the quality and speed of overall model learning by influencing the invariant network, allowing the model to converge more quickly.

4.4 The Impact of Equivariance on Invariance Loss

We believe that using invariant transformed samples in equivariability learning can facilitate the model to learn the invariant for the second time. In Fig. 2, which shows the invariance loss of each method, we can see that the invariance loss of our method(transformation integration) is the lowest, which proves that equivariance learning in our method can indeed help invariance learning, and invariance learning in equivariance learning in another perspective, learning to acquire semantic features that are difficult to obtain in invariant learning.

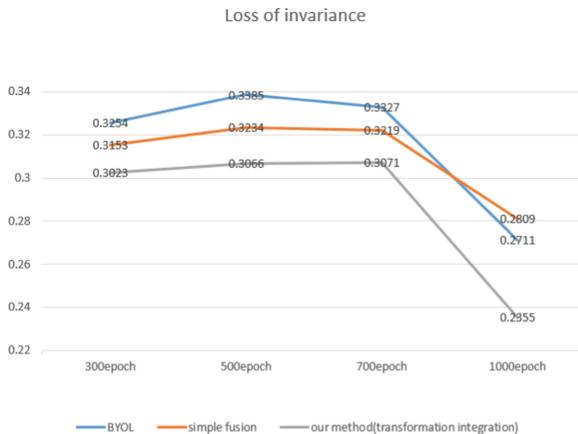


Fig. 2. CIFAR-10: Invariance loss at different epochs using different methods

4.5 Metrics of Equivariance and Invariance

In Fig. 3 we show the impact of our approach on the equivariance and invariance of the transformations, we use negative cosine similarity to measure invariance, the lower this value the more similar the two views are, representing that the model can recognise the same class of views independently of the invariant transformations, encouraging invariance. Similarly, equivariance is measured using average cosine similarity. We compare six pairs formed from four rotated views, and a lower value indicates lower similarity among these views. This allows the model to better distinguish equivariant transformations, thus encouraging equivariance. We can see that both equivariance and invariance we show a better metric than other methods.

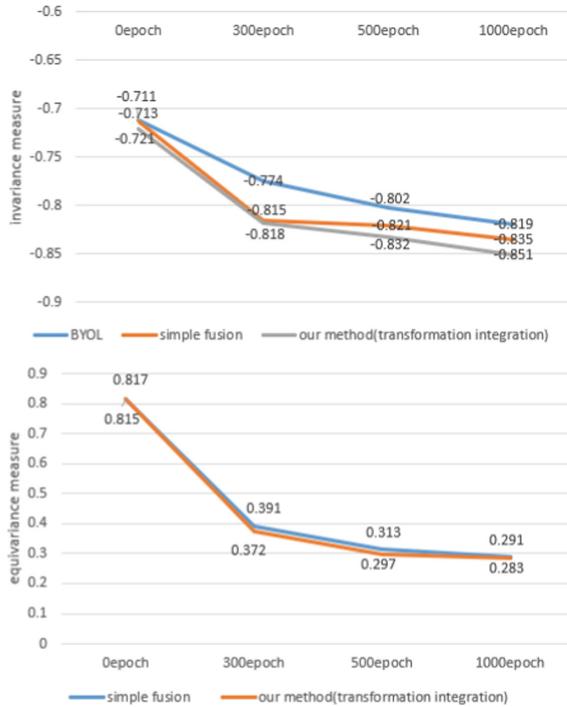


Fig. 3. CIFAR-10: Metrics of equivariance and invariance in the training process

4.6 Ablation Studies

In the ablation study, we compared the base model with the simple fusion of equivariance and invariance, as well as our method (transformation integration). In contrast, our method involved feeding the BYOL [17]-enhanced views into the equivariant network. We posit that rotating the augmented samples to determine the rotation angle not only assists the encoder in learning image features but also facilitates better learning of the transformations introduced by augmentation. As depicted in Fig. 4, our method outperformed the simple fusion, achieving superior results, with improvements also observed in KNN accuracy [31].

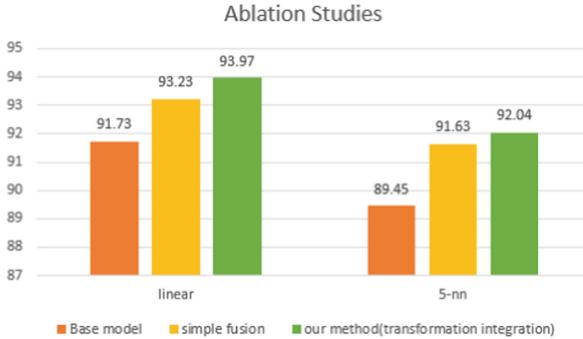


Fig. 4. CIFAR-10: accuracy of using different methods, trained for 1000 epochs

5 Conclusion

In this paper, we integrate equivariance and invariance, which differs from traditional self-supervised learning methods that solely rely on invariant transformations. Importantly, our integration is not a simple combination; the equivariance network can directly influence the invariance network, undergo secondary learning, and acquire semantically rich features, thereby accelerating model convergence. This integration achieves better complementarity between equivariance and invariance than previous approaches. Moreover, our method does not require modifications to the BYOL model, facilitating its transferability to other models. Through empirical validation, we demonstrate that our approach yields improved accuracy, surpassing the conventional benchmarks in computer vision.

References

1. Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 37–45 (2015)
2. Arazo, E., Ortego, D., Albert, P., O’Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
3. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. *Adv. Neural Inf. Process. Syst.* **32** (2019)
4. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478* (2021)
5. Caron, M., et al.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9650–9660 (2021)
6. 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)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. *arxiv 2020. arXiv preprint arXiv:2003.04297* (2003)

8. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15750–15758 (2021)
9. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
10. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International Conference on Machine Learning, pp. 2990–2999. PMLR (2016)
11. Da, K.: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
12. Dangovski, R., et al.: Equivariant contrastive learning. arXiv preprint [arXiv:2111.00899](https://arxiv.org/abs/2111.00899) (2021)
13. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International Conference on Machine Learning, pp. 3015–3024. PMLR (2021)
14. Fetterman, A., Albrecht, J.: Understanding self-supervised and contrastive learning with bootstrap your own latent (BYOL). Untitled AI, August (2020)
15. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8059–8068 (2019)
16. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint [arXiv:1803.07728](https://arxiv.org/abs/1803.07728) (2018)
17. Grill, J.B., et al.: Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural. Inf. Process. Syst.* **33**, 21271–21284 (2020)
18. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
19. Guo, H., Ba, Y., Hu, J., Si, L., Qiang, W., Shi, L.: Self-supervised representation learning with meta comprehensive regularization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 1959–1967 (2024)
20. 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)
21. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR (2015)
22. Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1413–1421 (2015)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. *CS 231N* **7**(7), 3 (2015)
25. Lenc, K., Vedaldi, A.: Understanding image representations by measuring their equivariance and equivalence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 991–999 (2015)
26. Loh, C., Christensen, T., Dangovski, R., Kim, S., Soljačić, M.: Surrogate-and invariance-boosted contrastive learning for data-scarce applications in science. *Nat. Commun.* **13**(1), 4223 (2022)

27. Metzger, S., Srinivas, A., Darrell, T., Keutzer, K.: Evaluating self-supervised pre-training without using labels. arXiv preprint [arXiv:2009.07724](https://arxiv.org/abs/2009.07724) (2020)
28. Peng, X., Wang, K., Zhu, Z., Wang, M., You, Y.: Crafting better contrastive views for siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16031–16040 (2022)
29. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)
30. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: self-supervised learning via redundancy reduction. In: International Conference on Machine Learning, pp. 12310–12320. PMLR (2021)
31. Zhang, H., Berg, A.C., Maire, M., Malik, J.: SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 2, pp. 2126–2136. IEEE (2006)
32. Zhang, L.: Equivariance and invariance for robust unsupervised and semi-supervised learning (2020)

Demonstration Paper



FedPPQs: Optimizing Property Path Queries Evaluation over Federated RDF Systems

Jibing Wu, Ningchao Ge^(✉), Tengyun Wang, Xuan Li, Lihua Liu,
and Hongbin Huang

Laboratory for Big Data and Decision, National University of Defense Technology,
Changsha, China

{wujibing,geningchao,wangtengyun18,lihualiu,hbhuang}@nudt.edu.cn

Abstract. Property path queries are a powerful type of query in SPARQL1.1 that enable users to perform conditional path searches similar to regular expression queries in SQL. However, there are still few federated RDF systems that can support property path queries. To address these challenges, in this demo, we propose and implement a federated RDF system named **FedPPQs** that can support property path queries based on MinDFA and B-DFS. **FedPPQs** transforms property path queries into an automaton matching problem and utilizes a fast matching method of MinDFA based on B-DFS to improve query efficiency. Experimental results demonstrate that **FedPPQs** achieves significant improvements in query efficiency.

1 Introduction

Knowledge graph has been widely studied by virtue of its powerful knowledge representation ability. Among that, RDF is usually used as the data organization form of knowledge graph. In recent years, a large number of RDF datasets have been published. These RDF datasets are usually managed by their publishers alone, and the sharing between datasets is insufficient. Therefore, researchers have proposed the federated distributed RDF system. In order to apply the centralized RDF system data query language SPARQL to the federated RDF system, scholars have done a lot of related query optimization work [1–4]. However, most of these works are aimed at SPARQL1.0 or special query types, and the optimization of property path query in SPARQL1.1 is still insufficient. To this end, This paper constructs a federated distributed RDF system named **FedPPQs**, which can effectively improve the efficiency of property path query. **FedPPQs** has the following features:

- **FedPPQs** constructs MinDFA by utilizing the regular features of property path expressions, and adopts isomorphic reuse strategy to reduce the construction cost of MinDFA.

- FedPPQs combines the characteristics of DFS and BFS, and adopts a fast matching method based on B-DFS to improve the matching efficiency of MinDFA.
- FedPPQs shows excellent performance by experimental evaluation.

2 System Architecture

Figure 1 gives the system architecture of federated RDF system FedPPQs. For a federated property path query submitted by a user, its query execution process includes three stages: query decomposition and RDF sources localization, MinDFA construction and MinDFA matching base on B-DFS.

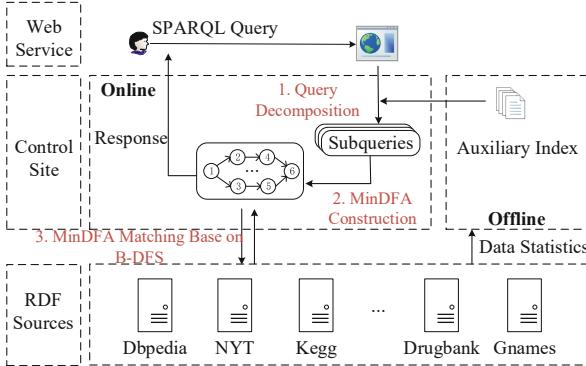


Fig. 1. The system architecture of federated RDF system FedPPQs

Query Decomposition and RDF Sources Localization. Given a federated RDF system $F = (C, S, d)$, where C is the control site, S is the set of RDF sources, and d is the mapping of vertices and edges in RDF graph to RDF source. In the offline stage, we count the meta information of subjects, predicates and objects of each RDF source to form an auxiliary index $PS_{Cardinal}$. $PS_{Cardinal} = \{< P_i, \{d(P_i), S(P_i), O(P_i)\} >, \dots, < P_i, \{d(P_i), S(P_i), O(P_i)\} >\}$, where P_i is a predicate of the federated RDF system F , $d(P_i)$ is the RDF sources set of P_i , $S(P_i)$ and $O(P_i)$ is the subject coefficient and object coefficient of P_i , respectively. The predicates of triples of a SPARQL federated query are usually constant, so we can quickly get their RDF sources' location through the constant predicate and the auxiliary index. Finally, we combine the constant predicates with the same single RDF source into a subquery. In particular, the triple whose predicate is a property path expression is regarded as a subquery alone.

MinDFA Construction. For subqueries containing regular triples, there are already very mature methods to execute them, so here we focus on property

path subqueries. Considering the similarity between property path expression and regular expression, we propose to construct an property path expression as a MinDFA, thus transforming the query problem into the automaton matching problem. The construction process of MinDFA mainly includes two steps: NFA construction based on Thomson, DFA construction and minimization based on subset construction and equivalent state merging. In addition, we propose the isomorphic reuse strategy to reduce the construction cost of MinDFA for multi-property path queries.

MinDFA Matching Based on B-DFS. Automata matching process of federated RDF system is the path search process based on graph. Breadth-first search (BFS) and depth-first search (DFS) are two typical graph search algorithms. However, the original BFS algorithm and DFS algorithm can not give full play to the distributed parallel processing ability of the federated RDF system, and BFS will bring extra memory consumption due to too many times of queues. Therefore, we propose a fast matching method of MinDFA based on B-DFS, as shown in Fig. 2. This method first performs a layer of width search, and then performs the deep search in parallel processing, which can effectively avoid memory overrun and improve the utilization rate and matching efficiency of distributed resources.

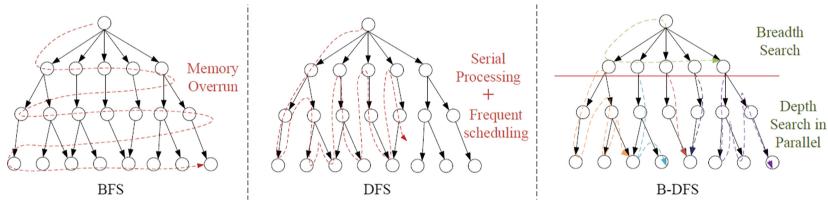
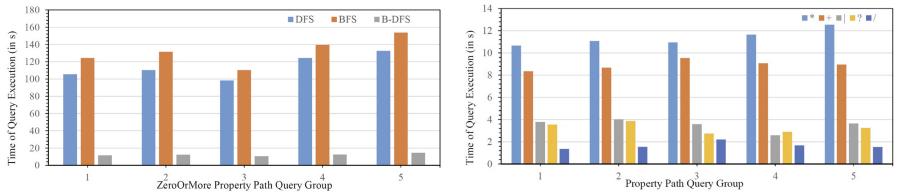


Fig. 2. The comparison of three matching strategies



(a) The performance of different matching strategies of MinDFA.

(b) The performance comparison of five property path query symbols under the same scale dataset.

Fig. 3. The results of effectiveness analysis of the proposed method

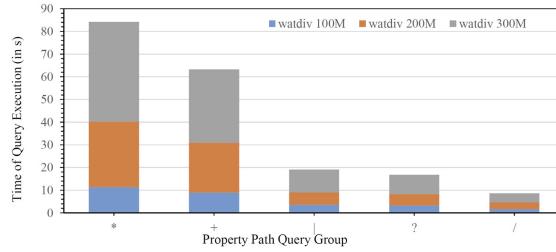
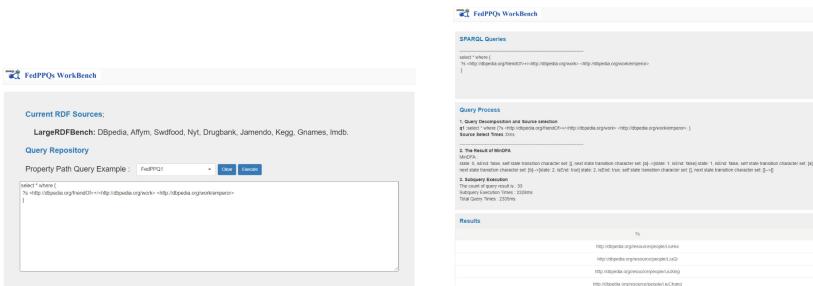


Fig. 4. The results of robustness analysis of the proposed method

3 Experiments and Demonstration

In order to verify the effectiveness of the proposed method, we implemented a federated RDF system, named **FedPPQs**. FedPPQs consists of six servers, one of which is the control site and the other are the RDF sources. The datasets include real dataset **LargeRDFBench** and synthetic dataset **WatDiv**. Figure 3(a) shows that the fast matching strategy based on B-DFS can effectively improve the matching efficiency of MinDFA. Figure 3(b) and Fig. 4 show the performance comparison and robustness analysis of five property path query symbols.



(a) The homepage of federated RDF system FedPPQs. **(b) The query result page of federated RDF system FedPPQs.**

Fig. 5. Main pages of federated RDF system FedPPQs

Figure 5(a) is the homepage of federated RDF system **FedPPQs**, which gives the RDF sources of system and property path query examples list. Users can select a given query example or enter the property path query statement they want to query in the query input box. Figure 5(b) shows the query result page of a property path query. On this page, users can obtain the complete execution process of the whole query, including the subqueries set after query decomposition, MinDFA generation result, query execution times and query results.

4 Conclusion

In this paper, an optimization method of property path query for federated distributed RDF system is proposed, and a prototype system **FedPPQs** is constructed. **FedPPQs** provides a friendly human-computer interaction interface and mainstream RDF datasets, which can effectively handle federated property path queries submitted by users.

Acknowledgement. This work was supported by the National Foundation Project for Postdoctoral Researchers of China (GZC20233528).

References

1. Ge, N., Qin, Z., Peng, P., Li, M., Zou, L., Li, K.: A cost-driven top-k queries optimization approach on federated RDF systems. *IEEE TBD* **9**(2), 665–676 (2023). <https://doi.org/10.1109/TBDDATA.2022.3156090>
2. Montoya, G., Skaf-Molli, H., Hose, K.: The *odyssey* approach for optimizing federated SPARQL queries. In: d’Amato, C., et al. (eds.) *ISWC 2017. LNCS*, vol. 10587, pp. 471–489. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_28
3. Peng, P., Ge, Q., Zou, L., Özsü, M.T., Xu, Z., Zhao, D.: Optimizing multi-query evaluation in federated RDF systems. *TKDE* **33**(4), 1692–1707 (2019)
4. Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: optimization techniques for federated query processing on linked data. In: *ISWC*, pp. 601–616 (2011)



MPCPM: Multi-level Prevalent Co-location Pattern Miner

Vanluan Nguyen and Vanha Tran^(✉) 

FPT University, Hanoi 155514, Vietnam
luannv@fpt.edu.vn, hatv14@fe.edu.vn

Abstract. Prevalent co-location pattern (PCP) mining are crucial in fields like medicine, biology, and urban planning to identifies spatial relationships between objects or their simultaneous occurrences. However, challenges such as dealing with heterogeneous spatial data and high computational costs for mining multilevel PCPs persist. This demonstration presents MPCPM (Multi-level Prevalent Co-location Pattern Miner), a system for users who are not only interested in co-location patterns and their levels but also in the high performance of mining SCPs. Users give a spatial data set, the designed miner evaluate and identifies SCPs which are global and local. We evaluate and identify the prevalent co-location patterns which are global and local. Additionally, MPCPM also cites and represents the sets of instances that make up the PCP to provide more information and help the decision making.

Keywords: Spatial co-location pattern · Multilevel PCP · POI datasets

1 Introduction

The constant evolution of technology leads to the daily collection of vast spatial data. Prevalent co-location pattern (PCP) mining, crucial in this context, identifies sets of events occurring together in space frequently [2], offering insights applicable to diverse fields like urban planning [6], public health [1], logistics [4].

For example, Fig. 1 illustrates the distribution of a spatial data set with many objects (called spatial instances), these objects are classified into four categories (called spatial features), i.e., {Restaurant}(A), {Shopping mall}(B), {Drinking shop}(C). As can be seen that, the instances of a group {A, B}, {A, C}, {B, C} and {A, B, C} are frequently occur together in the neighbor areas of each other, called a SCP and the size of the patter is 3 since it has three distinct features. Yet, analyzing co-location patterns encounters challenges due to uneven real-world data distribution. Recent studies [3, 5, 7] applying adaptive distance thresholds are effective in finding SCPs, but when applied to data with a large number of features, it is ineffective. To overcome these limitations, we propose a three-stage framework: Partitioning the dataset into subsets based on both density levels and distribution. Identifying maximal cliques within each cluster to detect candidate co-location patterns. Evaluating candidate patterns



Fig. 1. Spatial co-location patterns (SCPs).

using the Participation Index for prevalence measure and the Global Index to distinguish prevalent Global pattern from Local pattern.

In this paper, we develop a Multi-level Prevalent Co-location Pattern Miner (MPCPM) system for users, which take dominant relationship between the features that constitute co-location patterns and demonstrate their hierarchical structure. Given a set of spatial data (e.g., urban POI data), we aim to find MPCPs with global and local levels. We firstly discover prevalent co-location patterns, then identify levels from prevalent patterns. At last, the system will provide a visual analysis.

2 System Overview

As Fig. 2 illustrates, our system consists of three main components: (1) handles data processing, (2) the MPCP mining processor, and (3) the extraction and

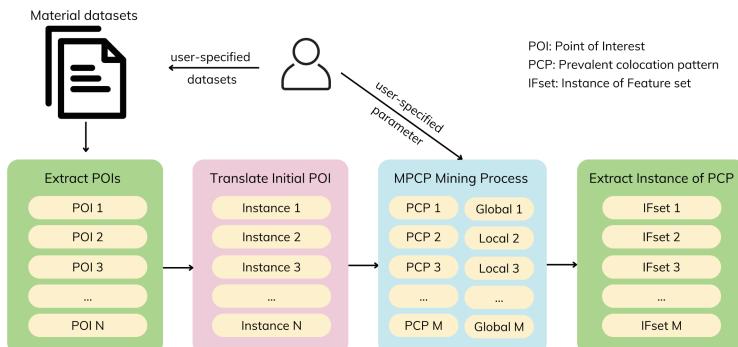


Fig. 2. The process of finding MPCP and visualization PCP levels.

visualization process. MPCP uses coordinate-based data (e.g., urban POI data) as initial input data.

In this demonstration, we collected POI datasets from user-specified file (primarily urban point-based data) obtained from public dataset as the initial data. In the handles data processing, we extract information about the set of locations considered as POIs, then classify and transform the fields of the initial POI data into the general input format of co-location mining as Feature type, Instance ID, coordinates to instantiate each POI as an input for MPCP mining. In the MPCP mining processor, users set parameters including the number of k neighbors, density level, prevalent threshold, and MLCP threshold based on their understanding of the data characteristics. We then determine the neighborhood relationships between instances by the K nearest neighbors, partition the data regions based on different densities, and mine the PCP (prevalent co-location patterns) using the Participation Index. Finally, we determine the level of each PCP based on the MLCP threshold. In the extraction and visualization process, we compile important information about the data, save the lists of global PCPs, local PCPs, and the set of instances of features during the mining process. This information is visualized in the user interface, displaying the specific distribution of relationships between the features of each PCP on a coordinate system.

3 Demonstration Scenarios

In the MPCPM system, we developed an easy-to-use user interface with public POI datasets from Shenzhen to demonstrate the capabilities of MPCPM.

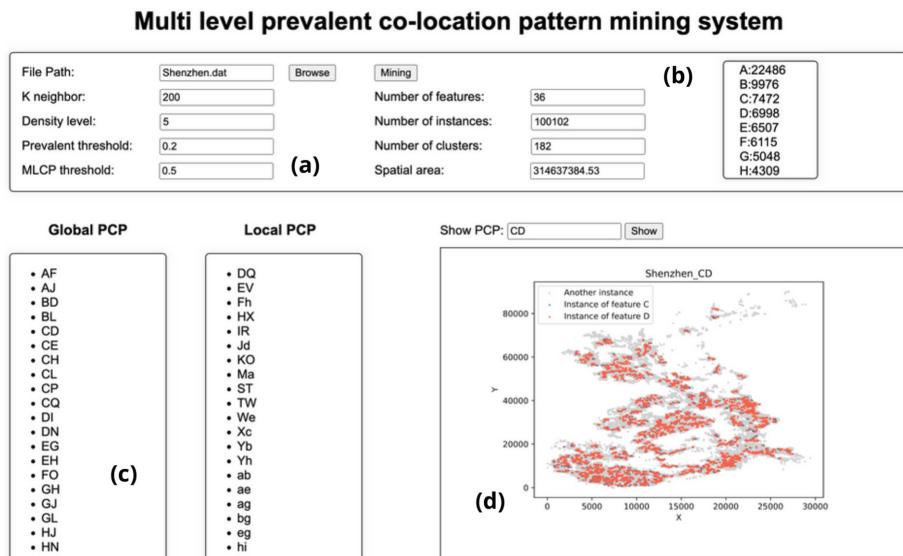


Fig. 3. The GUI of MPCPM in the processing and visualization.

As shown in Fig. 3, the “Multi-level prevalent co-location pattern mining system” provides a comprehensive set of functionalities organized into different sections of the interface. The Parameter Input Panel allows users to input various parameters necessary for the mining process, such as the file path for the data set, the number of neighbors (K neighbor), the density level, the prevalent threshold, and the MLCP threshold. Users can also browse the data file and start the mining process using the provided buttons.

Figure 3(b) shows when the mining process is completed we show the data summary panel displays a summary of the data set, including the number of features, instances, clusters, and the spatial area covered. Additionally, it provides a list of the counts of different features present in the dataset, such as A, B, C, etc., along with their respective counts.

As shown in Fig. 3(c), the Global and Local PCP Panels present the Global prevalent Co-location Patterns, listing patterns like {A,F}, {C,D}, {C,P}, etc., and the Local PCP, which includes more detailed patterns like {D,Q}, {E,V}, {S,T}, etc., helping users understand both overarching and specific patterns within the data. The Pattern Visualization Panel offers a tool for visualizing patterns.

As shown in Fig. 3(d), users can select a specific PCP from a dropdown menu and visualize its spatial distribution on a map, as shown in the example for the pattern {C,D} which displays instances of feature C and feature D across the Shenzhen area. This visualization helps users understand the spatial relationships and clustering of different features. Overall, this system allows for detailed input, comprehensive data summarization, clear presentation of both global and local patterns, and visualization of specific patterns within a geographic area.

4 Conclusions

In this demonstration, we developed a system to identify MPCPs to uncover the hierarchical relationships within prevalent co-location patterns. The demonstration scenarios highlighted the efficiency of our system. The MPCPM, utilizing POI data, illustrates its practical importance and can be further applied in several areas such as urban planning and commercial site recommendations.

References

1. Hou, W.: Applications of big data technology in intelligent transportation system. *Highl. Sci. Eng. Technol.* **37**, 64–71 (2023)
2. Huang, Y., Shekhar, S., Xiong, H.: Discovering colocation patterns from spatial data sets: a general approach. *IEEE Trans. Knowl. Data Eng.* **16**(12), 1472–1485 (2004)
3. Joo, S.Y., Shekhar, S.: A joinless approach for mining spatial colocation patterns. *IEEE Trans. Knowl. Data Eng.* **18**(10), 1323–1337 (2006)
4. Kanavi, V.S.: Big data analysis and its application in different industrial domains. *Int. J. Res. Appl. Sci. Eng. Technol.* **10**(7), 4255–4257 (2022)

5. Liu, Q., Liu, W., Deng, M., Cai, J., Liu, Y.: An adaptive detection of multilevel co-location patterns based on natural neighborhoods. *Int. J. Geogr. Inf. Sci.* **35**(3), 556–581 (2020)
6. Mondal, S.P., Mondal, S.P., Adhikari, S.K.: Applications of big data in various fields: a survey. In: *Advances in Intelligent Systems and Computing*, pp. 221–233 (2023)
7. Qian, F., Chiew, K., He, Q., Huang, H.: Mining regional co-location patterns with kNNG. *J. Intell. Inf. Syst.* **42**(3), 485–505 (2013)



FGAQ: Accelerating Graph Analytical Queries Using FPGA

Yi Ding¹, Zhengyi Yang²(✉) , Shunyang Li², Liuyi Chen³, Haoran Ning², Kongzhang Hao², and Yongfei Liu¹

¹ Euler AI, Sydney, Australia
`{kino,fayer}@eulerai.au`

² The University of New South Wales, Sydney, Australia
`{zhengyi.yang,shunyang.li,haoran.ning,k.hao}@unsw.edu.au`

³ Hunan University, Changsha, China
`chenliuyi2021@126.com`

Abstract. Field-programmable gate arrays (FPGAs) have significant advantages in parallelism and energy efficiency over CPUs and GPUs and are widely deployed by many enterprises and cloud server providers nowadays. In this paper, we demonstrate FGAQ, an FPGA-based system for accelerating graph queries on massive graphs. FGAQ supports the two most fundamental types of graph queries, namely subgraph and path queries, and features 1) a CPU-FPGA co-designed framework, 2) a fully pipelined FPGA execution, and 3) reduced data transfer from FPGA's external memory. FGAQ provides a user-friendly interface and significantly improved performance. Performance evaluation shows that FGAQ outperforms the most popular graph database, Neo4j, by up to three orders of magnitude. The demo video can be found at https://www.youtube.com/watch?v=pEkzw_DOQYE.

Keywords: Graph Database · FPGA · Subgraph Queries · Path Queries

1 Introduction

Graph has been playing an increasingly important role in data management, with the prevalence of graph data in different application domains in recent years [2]. There are two fundamental types of graph queries in graph analysis [1], namely *subgraph queries* and *path queries*. Given a pattern graph q and a data graph G , subgraph queries aim to find all subgraph instances in G that are isomorphic to q . As for path queries, they navigate the graph to investigate the relations between a source vertex s and a target vertex t within k hops to find all paths. Subgraph and path queries are associated with a wide spectrum of applications in the areas of network & IT operations, finance, e-commerce, cybersecurity, bioinformatics, chemistry, social science, etc. [12, 13].

Considerable efforts are made in both industry and academia to develop efficient systems for subgraph and path queries [3, 4, 6, 9, 11]. However, almost all

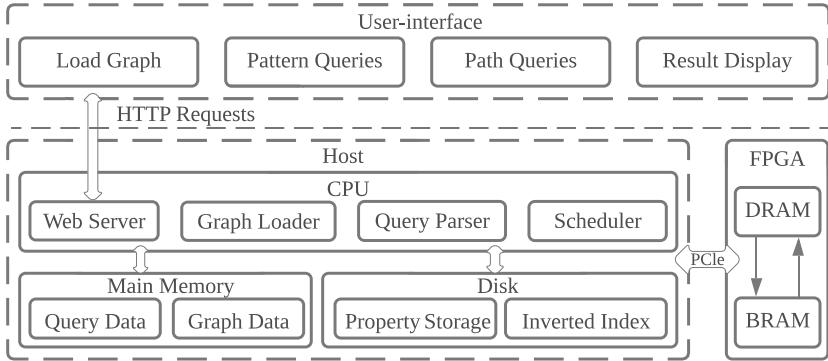


Fig. 1. System Architecture.

solutions are developed on CPUs, which have the following limitations when handling graph data: 1) CPUs do not offer flexible high-degree parallelism, and 2) CPU caches do not work effectively for irregular graph processing with limited data locality. With the recent advance of field-programmable gate arrays (FPGAs), people are provided with a new alternative to accelerate graph computations at the hardware level. FPGAs, which provide a new alternative to accelerate computation at the hardware level, have received much attention from researchers and enterprises. In terms of parallelism, FPGAs have shown significant advantages over multi-core CPUs due to their pipelining design and highly efficient hardware circuits.

Motivated by the above reasons and based on the latest research efforts [5, 7], we develop and demonstrate a prototype of FPGA-based Graph Analytical Query Engine (FGAQ). FGAQ supports subgraph queries and path queries. Specifically, FGAQ has the following features:

- CPU-FPGA co-design. FGAQ uses a CPU-FPGA co-designed framework, with the CPU handling parsing, preprocessing, and scheduling, while the FPGA performs computation for subgraph and path queries.
- Fully pipelined execution. The FPGA side is designed and implemented in a fully pipelined manner for massive parallelism and maximized efficiency.
- Reduced data transfer. As fetching data from FPGA's external memory (DRAM) is very expensive, FGAQ applies partition and caching techniques to reduce the data transfer between DRAM and BRAM.

2 System Design

Figure 1 shows the system architecture of FGAQ. On the frontend, FGAQ provides a user-friendly user interface for entering different types of queries, displaying query results and loading graph data from the local disk. Users are able to draw

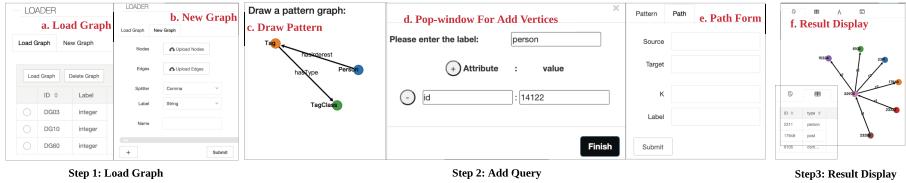


Fig. 2. Basic processing pipeline.

the subgraph and path queries when entering queries, and FGAQ will return and display the results either in a graph view or a table view. FGAQ uses *d3*¹ and *layui*² libraries to build the frontend. It communicates with the back-end web server on HTTP requests.

On the backend, FGAQ employs the CPU-FPGA co-designed framework. On the host side, FGAQ features a web server, a graph loader, a query parser, and a scheduler. The FPGA card is PCIe attached to the host, and the scheduler schedules the task on the host and FPGA and coordinates data transfer between them.

FGAQ supports property graphs [10] in which each vertex and edge can have arbitrary properties (i.e., any number of key-value pairs). We store the query and the topological structure of the data graph (together with the label information) in-memory (in compressed sparse row format) for fast access, and the properties in the data graph on disk using the most popular key-value store in Redis³. FGAQ can also build an additional inverted index of user-defined properties in the key-value store. As an interesting further work, we will investigate more complex index structures such as a B-tree.

Upon receiving a query, FGAQ will first parse the query (by the query parser), send it to the scheduler, and finally execute it on FPGA. The CPU can access the graph data (either from the main memory or the disk) and send the data to FPGA via PCIe bus when needed. The backend of FGAQ is implemented in C++, except the webserver, which is implemented in Python with the help of the *flask*⁴ library. On the FPGA side, we implement the algorithm for subgraph queries based on [5] and the path queries based on [7].

3 Demonstration Overview

The demonstration mainly presents: 1) the primary processing pipeline of FGAQ; 2) LDBC queries; and 3) real-life applications. Throughout the demonstration, the attendee will be able to get familiar with the system architecture of FGAQ as well as its competitive performance.

¹ <https://d3js.org/>.

² <https://wwwlayui.com/>.

³ <https://github.com/redis/redis>.

⁴ <https://flask.palletsprojects.com/>.

Processing Pipeline. In this section, we guide the attendees to experience the whole processing pipeline of FGAQ. The basic pipeline is shown in Fig. 2, which includes the following three steps: 1) Load/Import graph; 2) Draw patterns/paths; 3) Display results.

LDBC Queries. In this scenario, we will pre-load an LDBC dataset in the server and allow the attendee to specify one of the benchmark queries. The query will be executed using both FGAQ and Neo4j. The performance metrics will be delivered back to the scene and demonstrated to the attendee to show the performance advantages of FGAQ.

Performance Evaluation. We adopt the LDBC social network benchmark (SNB) [8] to evaluate the performance of FGAQ. SNB provides a data generator that generates a synthetic social network together with a set of queries. Two datasets are used in the demo, SF10 and SF60, in this demo. SF10 has 29.99 million vertices and 176.48 million edges. SF60 has 187.11 million vertices and 1.25 billion edges. These datasets are generated by simulating a real social network akin to Facebook with a duration of 3 years.

We compare FGAQ with the most popular graph database system Neo4j. For subgraph queries, we use four representative subgraph pattern in [4] selected from the complex workload of LDBC-SNB. For path queries, we vary the length constraint k from 2 to 5 (denoted as P_2 to P_5 , respectively), and set the edge label constraint to \emptyset . For each k , we randomly generate 10 query pairs $\{s, t\}$, and report the average time.

FGAQ significantly outperforms Neo4j on both subgraph and path queries in both datasets, achieving 100% completion rate where Neo4j only completes 56% of queries. For SF10, FGAQ achieves an average speedup of 13 times on subgraph queries (up to 105 times), and an average speedup of 1180 times on path queries (up to 2133 times) compared with Neo4j. For SF60, Neo4j is only able to complete one query among the four subgraph queries. FGAQ achieves a 772 times speedup on subgraph query, and an average speedup of 721 times on path queries (up to 1095 times).

4 Conclusion and Future Work

In conclusion, we demonstrate the FPGA-based prototype system for subgraph and path queries. In the future, we will work on supporting graph query language such as Cypher/GQL and/or integrating FGAQ into existing graph database systems to accelerate more graph queries.

References

- Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., Vrgoč, D.: Foundations of modern query languages for graph databases. *ACM Comput. Surv. (CSUR)* **50**(5), 1–40 (2017)
- Bonifati, A., Özsu, M.T., Tian, Y., Voigt, H., Yu, W., Zhang, W.: The future of graph analytics. In: Companion of the 2024 International Conference on Management of Data, pp. 544–545 (2024)
- Chen, K., Wen, D., Zhang, W., Zhang, Y., Wang, X., Lin, X.: Querying structural diversity in streaming graphs. *Proc. VLDB Endow.* **17**(5) (2024)
- Hao, K., Yang, Z., Lai, L., Lai, Z., Jin, X., Lin, X.: Patmat: a distributed pattern matching engine with cypher. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2921–2924 (2019)
- Jin, X., Yang, Z., Lin, X., Yang, S., Qin, L., Peng, Y.: Fast: FPGA-based subgraph matching on massive graphs. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1452–1463 (2021)
- Lai, L., et al.: Distributed subgraph matching on timely dataflow. *Proc. VLDB Endow.* **12**(10), 1099–1112 (2019)
- Lai, Z., Peng, Y., Yang, S., Lin, X., Zhang, W.: PEFP: efficient k-hop constrained ST simple path enumeration on FPGA. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1320–1331 (2021)
8. LDBC: LDBC benchmark (2024). <http://ldbcouncil.org/benchmarks>
- Peng, Y., Zhang, Y., Lin, X., Zhang, W., Qin, L., Zhou, J.: Towards bridging theory and practice: hop-constrained S-T simple path enumeration. *Proc. VLDB Endow.* 463–476 (2019)
10. Wang, R., Yang, Z., Zhang, W., Lin, X.: An empirical study on recent graph database systems. In: Knowledge Science, Engineering and Management, pp. 328–340 (2020)
11. Yang, Z., Lai, L., Lin, X., Hao, K., Zhang, W.: Huge: an efficient and scalable subgraph enumeration system. In: Proceedings of the 2021 International Conference on Management of Data, pp. 2049–2062 (2021)
12. Yu, J., et al.: Group-based fraud detection network on e-commerce platforms. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 5463–5475 (2023)
13. Zhu, G., Lin, X., Zhu, K., Zhang, W., Yu, J.X.: Treespan: efficiently computing similarity all-matching. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 529–540 (2012)



A Progressive Question Answering Framework Adaptable to Multiple Knowledge Sources

Yirui Zhan¹, Yanzeng Li², Minhao Zhang², and Lei Zou^{2(✉)}

¹ Sichuan University, Sichuan, China

zhanyirui@stu.scu.edu.cn

² Peking University, Beijing, China

liyanzeng@stu.pku.edu.cn, {zhangminhao,zoulei}@pku.edu.cn

Abstract. Existing deep learning-based models for knowledge base question answering (KBQA) suffer from the high costs of adapting the system to disparate datasets in real-world scenarios (e.g., multi-tenant platform). In this paper, we present ADMUS, a progressive knowledge base question answering framework designed to accommodate a wide variety of datasets with multiple languages by decoupling the architecture of conventional KBQA systems. Our framework supports the seamless integration of new datasets with minimal effort, only requiring creating a dataset-related micro-service at a negligible cost. To enhance the usability of ADUMS, we design a progressive framework consisting of three stages, ranging from executing exact queries, generating approximate queries and retrieving open-domain knowledge referring from large language models. An online demonstration of ADUMS is available at: <https://answer.gstore.cn/pc/index.html>.

Keywords: Knowledge Base Question Answering · Large Language Model

1 Introduction

A Knowledge Base Question Answering (KBQA) system aims to retrieve or query the correct answers from the Knowledge Base (KB) based on a given Natural Language Question (NLQ) [7]. The Semantic Parsing (SP) framework is a reliable solution for KBQA, which is to convert NLQ into logical structures to generate KB queries and retrieves results through graph database interfaces (e.g., SPARQL and Cypher) [2,3]. It demonstrates the capability to handle complex and multi-hop questions effectively. Therefore, it has been widely adopted by many state-of-the-art KBQA systems recently [4,10,15].

This work was done during the internship of Yirui Zhan at Peking University. Y. Zhan and Y. Li—Contributed equally to this work.

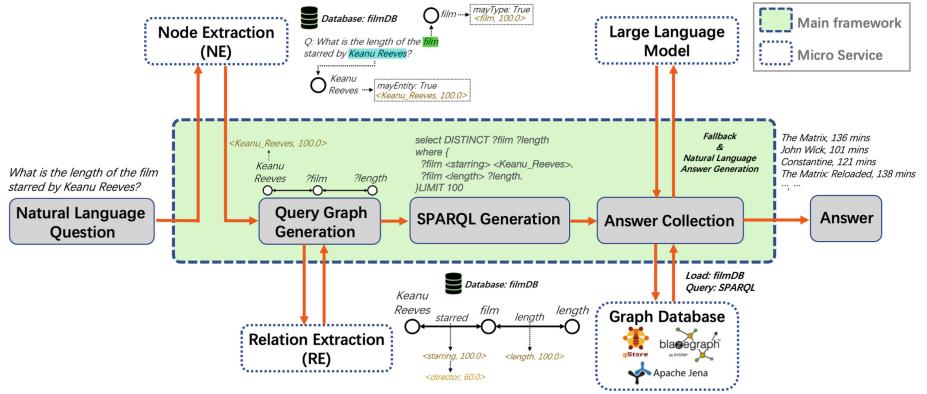


Fig. 1. Architecture of ADMUS.

However, the existing KBQA systems predominantly concentrate on improving performance metrics on benchmark datasets, disregarding the significant costs involved in re-training, re-deploying and adapting the system to different KBs and datasets (e.g., multi-tenant scenario). Therefore, we propose a flexible KBQA framework designed for seamless ADaptation to Multiple knowledge Sources, named ADMUS. ADMUS has the capability to accommodate multi-tenants with various user-provided language-agnostic KBs and QA datasets.

The main contributions of this demonstration can be summarized as follows: (1) We design ADMUS by decoupling dataset related services and dataset independent modules within the SP-based KBQA pipeline. This design adapts diverse KBs and has the potential for multi-tenant KBQA platforms. (2) We present a progressive framework on the top of ADMUS, starting from generating exact queries to approximate queries, ending with query with LLMs to ensure high usability. (3) To showcase the capabilities of our framework, we implement a web demonstration that incorporates several KBs.

2 Architecture

As shown in Fig. 1, the framework of ADMUS comprises dataset-related microservices and a dataset-independent backbone. Given an NLQ after selecting the dataset, the correlated Node Extraction (NE) service detects all entity mentions within the question. Then the Query Graph Generation module constructs a semantic query graph based on the entity mentions with semantic parsing techniques (e.g. Dependency Tree). Simultaneously, the edges are utilized by the Relation Extraction (RE) service to extract the relations within entity pairs. Then the complete query graph has been generated, and a subgraph matching strategy is employed to generate candidate SPARQL queries. Finally, the Answer Collection (AC) module executes SPARQLs in the corresponding KB. If none of the SPARQLs yields a result, the AC module will invoke an external LLM

with prompted query as a fallback. In the following paragraphs, we will provide a detailed description.

Node Extraction service identifies nodes within the query graph based on the input NLQ. There are four types of nodes: *Entity*, *Type*, *Literal*, and *Variable*. For entities and types, the NE module aims to extract their mentions from the input question via NER technologies [8]. The extracted mentions are then mapped to a set of entity (type) names in the knowledge graph using dataset-specific entity linking methods [12]. And several linking methods are adopted. For small KBs, linking entities can be achieved through substring matching or neural-linker approaches [11]. For large KBs like DBpedia, offline dictionary-based linker [3] or third-party entity linking services (e.g., DBpedia Lookup tool¹) are commonly adopted. Meanwhile, the literal attributes and the mention of variables would be extracted, refined and then merged with linked entities and types to construct the draft structure.

Query Graph Generation module begins with the skeleton parsed from the semantic structure of NLQ, and arranges nodes which are extracted from the NE service. The process of building the query graph involves determining whether there is an edge connecting every pair of nodes. Commencing with a special variable node (namely “target node”), we employ a depth-first search algorithm to identify which nodes it is connected to. This is done by recursively traversing the whole semantic structure around the target node until other nodes belonging to the query graph are encountered. These traversed nodes are then assumed to be connected to the target node.

Relation Extraction service identifies predicates that connect each pair of nodes in the query graph. Given the query graph as input, the dataset-related RE service will generate the relation (aligning with predicates in KB) for node pairs along with their corresponding scores. According to the size and complexity of the target KB, RE services employ various dataset-related methods (e.g., predicate dictionary [14], deep learning-based method [9], etc.) to identify all pairs of related nodes and their associated predicates.

SPARQL Generation module. ADMUS will process an approximate subgraph matching algorithm to search for corresponding subgraphs within the target KB (1) An entity node in the query graph corresponds to an entity node in the target KB. (2) A type node in the query graph corresponds to a type node in the target KB or an entity node that belongs to that type. (3) A variable node is treated as a wildcard, allowing it to potentially map to any node in the target KB. Once k subgraphs have been successfully matched, k SPARQL queries are generated.

¹ <https://github.com/dbpedia/dbpedia-lookup>.

Answer Collection module. In this module, an external LLM service is leveraged for activating conversational QA and responding to open-ended questions. The correctly retrieved responses from the database engine will be collected and rewritten based on an exquisitely designed prompt template. For any failed queries, the LLM will provide fallback answers to maintain a user-friendly experience, though the accuracy of these fallback responses is not guaranteed [6].

3 Demonstration and Discussion

We have utilized various datasets of different scales and languages to demonstrate our frameworks' capabilities, as presented in Table 1. To showcase the SP-based KBQA system for multiple KB sources that can be easily switched, we have developed a web interface for ADMUS, as illustrated in Fig. 2. This interface provides a comprehensive visualization of the entire process of ADMUS. It also offers the functionality to switch between different KB sources and utilize the LLM fallback feature.

Table 1. Datasets for Demonstration.

Name	Language	Triples	Entities	Predicates
birdDB	Chinese	17,607	10,704	14
filmDB	Chinese	4,531,096	437,986	32
DBpedia2016	English	198,969,616	8,465,000	59,486

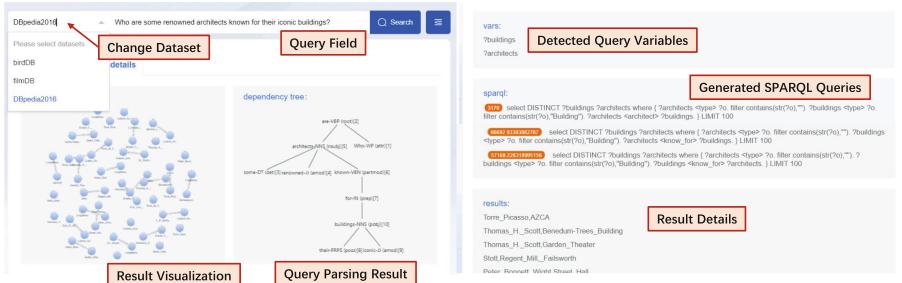


Fig. 2. Screenshot of ADMUS.

When a user wants to integrate a new NE service, there are various methods available, such as entity mention dictionaries [3], neural network models, or few-shot NER approaches [5]. Similarly, when a user incorporates a new RE service, predicate dictionaries, Relation Detection [9], or some of few-shot models [13]

can be employed. Compared to end-to-end KBQA models, ADUMS can support a new dataset by adding NE and RE services with negligible cost rather than completely building a new model and annotating massive training data.

In summary, our proposed progressive framework allows for exact answers through SP-based KBQA while leveraging the open-domain capabilities of LLMs as a fallback. Within various components of ADMUS, LLMs can be used to enhance their capabilities. For example, by using LLMs with prompts, it is possible to achieve few-shot NE and RE models, thereby strengthening the NE and RE modules in ADMUS and implementing a few-shot KBQA system. From the other side, ADUMS can serve as a plugin in LLM manners (e.g., LangChain [1]), enabling it to function as a personalized KBQA server to answer domain-specific questions exactly.

References

1. Chase, H.: LangChain, October 2022. <https://github.com/hwchase17/langchain>
2. Gu, Y., Pahuja, V., Cheng, G., Su, Y.: Knowledge base question answering: a semantic parsing perspective. arXiv preprint [arXiv:2209.04994](https://arxiv.org/abs/2209.04994) (2022)
3. Hu, S., Zou, L., Yu, J.X., Wang, H., Zhao, D.: Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Trans. Knowl. Data Eng.* **30**(5), 824–837 (2017)
4. Hu, X., Shu, Y., Huang, X., Qu, Y.: EDG-based question decomposition for complex question answering over knowledge bases. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 128–145. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_8
5. Huang, J., et al.: Few-shot named entity recognition: a comprehensive study. [arXiv:2012.14978](https://arxiv.org/abs/2012.14978) (2020)
6. Ji, Z., et al.: Survey of hallucination in natural language generation. *ACM Comput. Surv.* **55**(12), 1–38 (2023)
7. Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W.X., Rong Wen, J.: A survey on complex knowledge base question answering: methods, challenges and solutions. [arXiv:2105.11644](https://arxiv.org/abs/2105.11644) (2021)
8. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **34**(1), 50–70 (2020)
9. Li, Y., Hu, S., Han, W., Zou, L.: CORD: a three-stage coarse-to-fine framework for relation detection in knowledge base question answering. In: Proceedings of the 32nd ACM International CIKM (2023)
10. Omar, R., Dhall, I., Kalnis, P., Mansour, E.: A universal question-answering platform for knowledge graphs. *Proc. ACM Manage. Data* **1**(1), 1–25 (2023)
11. Sevgili, Ö., Shelmanov, A., Arkhipov, M., Panchenko, A., Biemann, C.: Neural entity linking: a survey of models based on deep learning. *Semantic Web* (Preprint), 1–44 (2022)
12. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.* **27**(2), 443–460 (2014)
13. Wen, W., Liu, Y., Ouyang, C., Lin, Q., Chung, T.: Enhanced prototypical network for few-shot relation extraction. *Inf. Process. Manage.* **58**(4), 102596 (2021)

14. Xue, B., Hu, S., Zou, L., Cheng, J.: The value of paraphrase for knowledge base predicates. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9346–9353 (2020)
15. Zhang, M., Zhang, R., Li, Y., Zou, L.: Crake: causal-enhanced table-filler for question answering over large scale knowledge base. [arXiv:2207.03680](https://arxiv.org/abs/2207.03680) (2022)



RocolSys: An Automatic Row-Column Data Storage System for HTAP

Shuangshuang Cui¹, Hongzhi Wang^{1(✉)}, Hao Wu¹, Dong Wang¹, Jinxuan Li¹, Jingbiao Ren², Chenguang Li², and Wei Zhao²

¹ Faculty of Computing, Harbin Institute of Technology, Harbin, China

{cuishuang, wangzh}@hit.edu.cn

² GBASE, Tianjin, China

Abstract. Hybrid Transactional and Analytical Processing (HTAP) brings great challenges to data storage systems. However, traditional approaches have struggled to dynamically adapt storage structures to data and complex workloads. Fortunately, machine learning can provide new methods to guide decisions on data storage structure selection. Motivated by this, we develop RocolSys, an automatic hybrid data storage system for HTAP that can predict workloads and select storage structure automatically based on machine learning. RocolSys predicts workloads accurately and selects storage structures efficiently. It also provides the user-friendly interface that allows users to connect to their own databases. We demonstrate the efficiency of RocolSys on public benchmarks.

Keywords: row and column storage · storage structure selection · workload forecasting · machine learning

1 Introduction

With the development of big data, modern enterprise-class database systems need to support HTAP. HTAP relies on a single system to process the mixed workloads of transactions and analytical queries simultaneously [1]. For the storage structure, row-based storage is suitable for highly concurrent transactional queries, while column-based storage is beneficial for analytic query processing. In order to support both high concurrency and high real-time workloads, the database needs to adopt a row and column coexistence storage structure.

Existing data storage methods support the row storage, column storage or mixed row and column storage [2]. However, the data storage structure mainly depends on the database administrators (DBAs). Fortunately, machine learning techniques have been used in recent years to address automatic data management techniques [3], which can provide prediction and decision-making services. Thus, we develop RocolSys, an automatic Row-column data storage System For HTAP based on machine learning, which focuses on the following objectives:

(1) Automatic storage selection. Given the data schema and the workload of the data set, RocolSys can recommend an efficient storage structure for the database based on machine learning, i.e., row-based or column-based storage.

(2) Efficient workload forecasting. RocolSys can predict the workload in the future according to the historical query Logs. It supports dynamic adjustment for the storage structure to adapt to future workload changes.

(3) User personalized selection. RocolSys can show users the execution time of different storage schemes corresponding to the workload. In addition, it provides a variety of time series prediction models. Users can flexibly choose the prediction model and storage structure according to the actual situation.

2 System Overview and Implementation

Figure 1 shows the architecture of RocolSys, including *User Interaction*, *Storage Structure Selection* and *Workload Forecasting*. Considering sufficient storage space, we chose the OpenGauss [4] database as the backend since it has a row storage structure as well as a column storage structure. In addition, users can also replace the backend with their own row-column database.

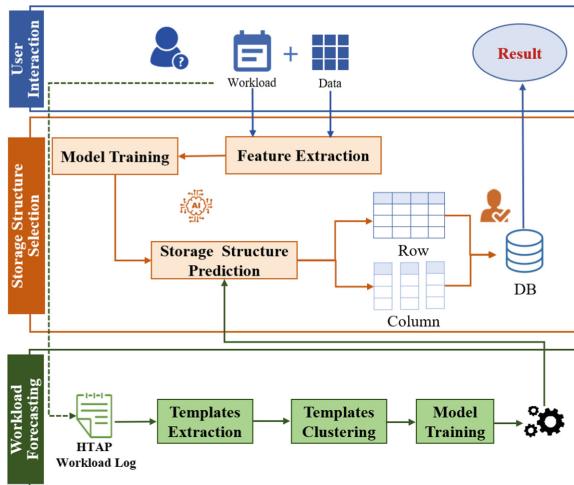


Fig. 1. System Architecture

User Interaction. RocolSys provides friendly interfaces for users to enter data, workloads, and queries. For demonstration purposes, users can compare the performance of machine learning methods with competitors in RocolSys.

Storage Structure Selection. After users upload the data and workload, RocolSys can recommend the storage structure. First, RocolSys can partition

large-scale data and automatically determine the storage structure. To achieve automatic storage, we developed an automatic selection algorithm for storage structure based on machine learning. RocolSys provides a user-friendly interface to determine the storage structure under the guidance of the system.

Workload Forecasting. This component is responsible for predicting future workload based on historical data. It helps users to cope with future workloads changes and adaptively adjust the data storage structure in advance. RocolSys provides a variety of pre-set time series prediction models, which can give the corresponding prediction accuracy results for users to choose.

3 Key Technologies

In this section, we introduce two key machine learning based methods in RocolSys.

Storage Structure Selection. To select the least cost storage structure for the user's data. Firstly, it needs to prepare the data, then divide the data and perform feature selection. RocolSys uses data schema and workload features, including *key field size*, *non-key field size*, *the number of fixed-length fields*, *the number of variable-length fields*, and *the number of rows* involved in a single operation. Then, RocolSys designs the storage decision cost model and uses the performance data to train this model. It analyzes the performance data collected above and designs the cost model. For a given workload and data schema S , the model calculates the cost of row and column storage respectively as shown below:

$$Cost_{row}(S) = W_1 * V_{row-insert}(S) + W_2 * V_{row-select}(S) \quad (1)$$

$$Cost_{column}(S) = W_1 * V_{col-insert}(S) + W_2 * V_{col-select}(S) \quad (2)$$

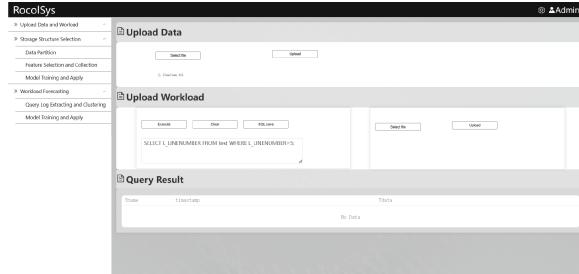
where W_1 denotes the number of queries inserted in the workload, W_2 denotes the number of queries found in the workload, and V_x denotes the predicted value corresponding to the x model ($x = \text{insert/select}$ model under row/column store). Finally, RocolSys uses XGBoost [5] to train regression models, which is not only efficient but also lightweight.

Workload Forecasting. To predict the workloads in the future based on historical workloads. The first step is to process the query logs. RocolSys extracts the SQL query from each line of the log file and replaced all numeric constants in the query's SQL string with $\$$. Then RocolSys optimizes DBSCAN to cluster the query templates. RocolSys specifies the distance between object and cluster center, as the evaluation criterion. The DTW distance function is then used as a similarity measure between the historical arrival rates of the two templates. Only when the similarity of query templates within each cluster is high, the prediction arrival rate can be more accurate when the prediction model is deployed for it. RocolSys supports six popular time series prediction algorithms and ensemble learning models to predict query arrival rates [6].

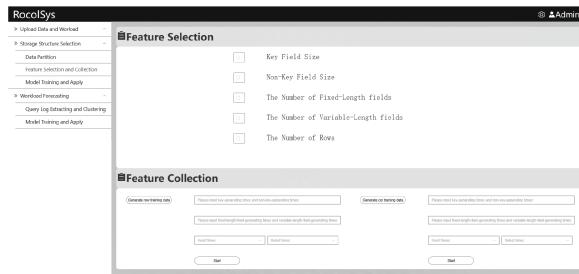
4 Demonstrations

We use the common benchmark set TPC-H [7] to demonstrate the data storage structure recommendation capabilities of Rocolsys.

Storage Structure Selection. Rocolsys allows users to upload data and workloads as shown in Fig. 2(a). After that, users can choose whether to partition the data or not. Users then choose to use the system in either pre-trained model mode or re-trained mode when using the system. They can interact with the system to perform feature extraction and collection on the data to generate the training set needed for the model as shown in Fig. 2(b). In particular, Rocolsys can train the model and display the accuracy of the model to the user, making it easy for the user to adjust and optimize the model parameters based on the model's accuracy. Rocolsys can select a storage structure for the given data. When the selection result is displayed to the user, they can decide whether to apply the recommended storage structure or not.



(a) Upload Data and Workload



(b) Feature Selection and Collection

Fig. 2. GUI of Rocolsys

Workload Forecasting. Rocolsys allows users to customize the selection of query logs for a certain period as historical data for workload prediction, and choose from seven different time series prediction models for training. Similarly, Rocolsys can show users the accuracy of different time series prediction models.

It can visually provide the user with prediction results. Based on the workload prediction results, the data that will be accessed frequently in the future can be identified. RocolSys will recommend column stores for this data. RocolSys can get around 85% performance improvement for users by the model of storage structure recommendation.

Acknowledgement. This work is supported by the NSFC 62232005, 62202126; the National Key Research and Development Program of China (2021YFB3300502) and China GBase Corp. Ltd.

References

1. Zhang, C., Li, G.L., Feng, J.H., Zhang, J.T.: Survey of key techniques of HTAP databases. *Ruan Jian Xue Bao/J. Softw.* **34**(2), 761–785 (2023)
2. Abadi, D.J., Madden, S.R., Hachem, N.: Column stores vs. row stores: how different are they really? In: Proceedings of the 2008 ACM SIGMOD (2008)
3. Domingos, P.: Machine learning for data management: problems and solutions. In: SIGMOD, pp. 629–629 (2018)
4. OpenGauss. <https://opengauss.org/zh/>
5. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD (2016)
6. Ma, L., Aken, D.V., Hefny, A., et al.: Query-based workload forecasting for selfdriving database management systems. In: The International Conference (2018)
7. TPC-H (2023). <https://www.tpc.org/tpch/>



MIPC-SHOPs: An Online System for Mining the Influence of Industrial Pollution on Cancer Based on the Spatial High-Influence Ordered-Pair Patterns

Lingli Zhang¹, Lizhen Wang²(✉), Peizhong Yang¹, and Lihua Zhou¹

¹ School of Information Science and Engineering, Yunnan University, Kunming 650091, China

² School of Science and Technology, Dianchi College, Kunming 650228, China

lzhwang@ynu.edu.cn

Abstract. Spatial ordered-pair pattern mining can be used to discover the potential spatial association between industrial emitted outdoor air pollutants and cancer cases. However, due to the significant correlation between the influence of pollution sources on cancer and the distance between them, and the influence of pollution sources by weather, emission concentration, and other factors is different, there are still some challenges in spatial ordered-pair pattern mining. In this paper, we design an online system MIPC-SHOPs for mining the influence of industrial pollution on cancer based on spatial high-influence ordered-pair patterns. Unlike previous works, first, a new influence measure based on Gaussian kernel density estimation is proposed in MIPC-SHOPs, which solves the problem that the influence of pollution sources on cancer cases decays with distance. Second, to restore the diffusion influence of pollution sources in the real world as much as possible, the urban wind direction, wind speed, and pollution emission concentration were considered to set a new spatial neighbor relationship measurement criterion. Considering the different carcinogenic levels of the pollution sources, a weighted method is proposed to calculate the influence of pollutant on cancer. In particular, the user can obtain the mining results through a simple interaction with the system.

Keywords: Online mining system · Spatial high-influence ordered-pairs · Interactive mining

1 Introduction

Spatial co-location pattern mining aims to find patterns where instances are frequently adjacent in geographical space [1]. These patterns may have important significance for understanding and predicting spatial phenomena. In the public health field, using the theory of spatial co-location pattern mining to analyze the association between cancer cases and pollution sources in geographical space can provide a scientific basis for environmental protection, public health and measures for reduce cancer risk. [2, 3] proposed a spatial high-influence co-location rule mining framework to extract high-influence co-location rules between pollution sources and cancer cases. [4] first proposed

the concept of spatial ordered-pair pattern, which can represent the influence relationship between pollution sources and cancer cases. The sources of pollution discussed in the above methods are all types of factories, and there are many types of pollutants emitted by a factory. Some of these substances have potent carcinogenicity, while some are not carcinogenic. The spread of these pollutants is affected by many realistic factors, such as wind direction, pollutant emission concentration, etc. The existing studies do not consider these factors. When measuring the influence of pollution sources on cancer cases, it is not in line with the reality that the degree of participation of pollution source features is simply accumulated.

In this paper, we created a system MIPC-SHOPs for mining the influence of industrial pollution on cancer based on spatial high-influence ordered-pair patterns. First, the pollution source was refined from “factory” to “carcinogen,” and the diffusion mechanism of the pollution source was modeled. Second, the kernel density estimation model is used to transforms the pattern interestingness metric into a density-based calculation method related to distance attenuation, called influence degree. Third, the efficiency of spatial ordered-pair pattern mining is improved based on the star materialized model. Finally, the system will provide a visual presentation.

2 System Overview

As shown in Fig. 1, the MIPC-SHOPs system mainly consists of four modules: (1) data input and visualization; (2) Modeling of pollution sources diffusion mechanism; (3) Mining spatial high-influence order-pair patterns; (4) Visualization of mining results.

In the first module, the input data set contains pollution source data and cancer case data, two groups of spatial data points containing location information. The system provides spatial visualization of pollution source data and cancer case data. The second module models the input pollution source data. (1) The position of the pollution source will shift under the action of the dominant wind direction. The offset distance of the pollution source is calculated by inputting the wind speed. (2) The three-level influence radius of pollution source instances is calculated according to the emission concentration of pollution source instances, and the spatial neighbor relationship between instances is calculated by combining the activity radius of cancer cases; (3) According to the types of carcinogens in pollution sources, set the three-level carcinogenic coefficient, and use the “carcinogenic coefficient” as the weight for weighted summation to calculate the influence of various pollutants on cancer cases. In the third module, star materialization of pollution source data and cancer case data after modeling is carried out, and the interest degree of a pattern is evaluated based on the proposed influence degree metric. The influence degree calculation includes three steps: the influence rate calculation of pattern features, the weighted influence rate calculation of distinguishing feature differences, and the influence degree calculation of a pattern. Then, the relationship between the influence degree and the given threshold is judged. If the influence degree is greater than or equal to the user-given threshold, it is a spatial high-influence ordered-pair pattern. In the last module, the system provides visualization of mining results.

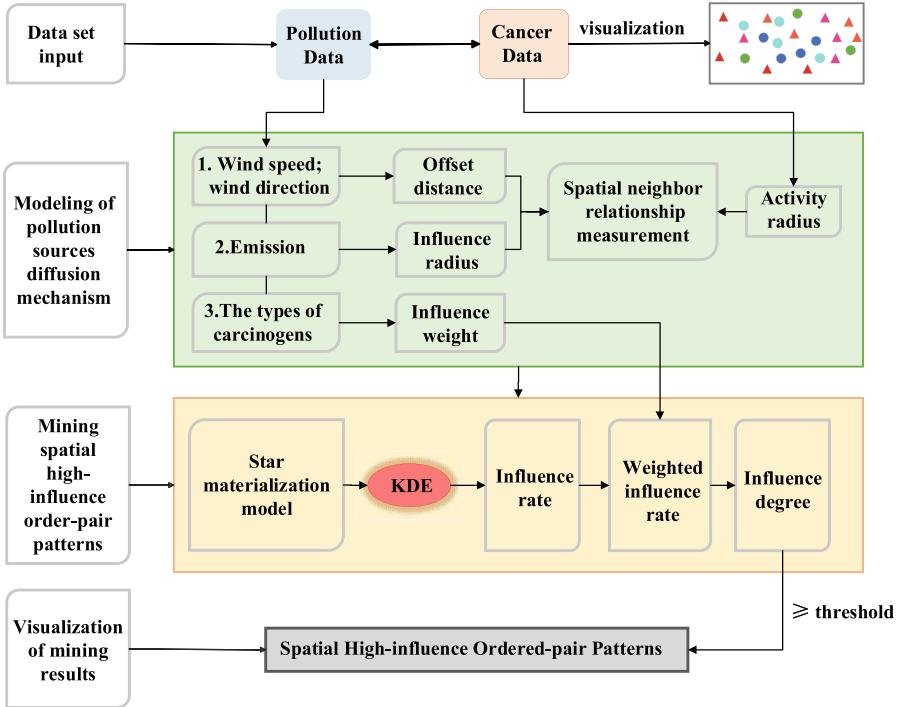


Fig. 1. The framework of MIPC-SHOPs.

3 Demonstration Scenarios

The MIPC-SHOPs system provides a good encapsulation and friendly interface. The cancer case data used in the system demonstration mainly comes from a hospital in Yunnan Province. The pollution source data are mainly formed by selecting pollutant discharge enterprise in the corresponding range according to the obtained cancer case data.

Figure 2 shows the user interface of the created system. Firstly, the user must upload the pollution source datasets and cancer case datasets to the system, which are located in the interface Fig. 2a. The system will display the spatial distribution of the uploaded data on the interface, as shown in Fig. 2b. Secondly, the relevant parameters need to be entered according to the results of the pollution source instance modeling. Figure 2c input wind speed v , range 0–15 m/s; At Fig. 2d, the three-level influence radius of the input pollution source is r_{min} , r_{mid} , r_{max} , and the cancer case activity radius of cancer cases r_c . At Fig. 2e, the third-level carcinogenic coefficient of input pollution source is $eps1$, $eps2$, and $eps3$, where $0 < eps1 < eps2 < eps3 < 1$. Figure 2f input smoothing factor k size, and $-1 < k < 1$, used to alleviate the problem of excessive deviation of impact rate caused by the difference in the number of cancer instance to assess the impact of pollution sources on cancer more accurately. Influence degree threshold min_prev of

patterns at Fig. 2g, $0 < \text{min_prev} \leq 1$. Finally, click the submission button in the position of Fig. 2h to perform spatial high-influence ordered-pair pattern mining.

Figures 2i and 2j are the result display interface of this system. Figure 2i shows the total number of features and instances of the input pollution source datasets and cancer datasets and outputs the number of spatial high-influence ordered-pair patterns. Figure 2j shows all the spatial high-influence ordered-pair patterns mined by the system. In order to facilitate users to view the patterns with a large influence degree intuitively, we sort the mining results in descending order of influence degree and show all the spatial high-influence ordered-pair patterns. For example, in the result, [AJ, a] is a high-influence ordered-pair pattern, “A” refers to particulate matter pollution, and “J” is soot pollution. “a” is lung cancer, indicating that {particulate matter, soot} has a close spatial correlation with lung cancer and has a significant impact on lung cancer.

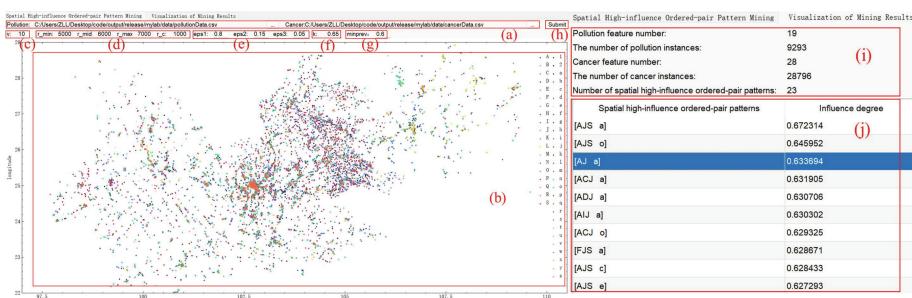


Fig. 2. Interface of MIPC-SHOPs and mining results display.

4 Conclusion

In this paper, we propose a framework for mining the influence of industrial pollution on cancer based on the spatial high-influence ordered-pair patterns and design an online system named MIPC-SHOPs for this purpose. The system mining efficiency increases by an average of 60% compared to other algorithms [4]. The mining results of MIPC-SHOPs can be widely used in urban planning, industrial pollution control, regional cancer screening, household site selection, and providing useful information for epidemiological experts.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (62276227, 62266050, 62306266), and the Yunnan Fundamental Research Projects (202201AS070015, 202401AT070450).

References

1. Wang, L., Fang, Y., Zhou, L.: Preference-based spatial co-location pattern mining. Springer Singapore (2022). <https://doi.org/10.1007/978-981-16-7566-9>

2. Lei, L., Wang, L., Zeng, Y., et al.: Discovering high influence co-location patterns from spatial data sets. In: 2019 IEEE International Conference on Big Knowledge (ICBK), pp. 137–144. IEEE, Beijing, China (2019)
3. Shu, J., Wang, L., Yang, P., et al.: Mining the potential relationships between cancer cases and industrial pollution based on high-influence ordered-pair patterns. In: Advanced Data Mining and Applications. ADMA 2022, pp. 27–40. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-22064-7_3
4. Xie, W., Wang, L., Chen, H., et al.: Identifying relationship between pollution sources and cancer cases with spatial ordered pair patterns. Data Anal. Knowl. Discov. **5**(2), 14–31 (2021)



A Perception System for DNS Root Service Status Based on Active and Passive Monitoring

Guozhong Dong, Hao Guo, and Hualong Wu^(✉)

Department of New Networks, Pengcheng Laboratory, Shenzhen 518055, China
{donggzh, guoh, wuhl}@pcl.ac.cn

Abstract. The Domain Name System's (DNS) root service serves as the backbone of the entire domain name resolution system. Its stability and reliability are pivotal to ensuring the smooth functioning of the network. To gain a real-time and comprehensive grasp of the root service's operational status, this paper introduces a perception system specifically designed for monitoring DNS root service status. This system integrates both active probing and passive monitoring techniques, providing a thorough and up-to-date understanding of DNS root resolution services within a designated network. This approach enables swift identification and resolution of any potential issues.

Keywords: Domain Name System · Root Service · Perception System · Active and Passive Monitoring

1 Introduction

The root service stands as a vital component of the Domain Name System, tasked with directing queries for top-level domains to their respective servers. Given its global significance and crucial role, the stability and performance of the root service have a profound influence on the overall network's functionality. As the Internet continues to expand and its complexity increases, the stability and performance of the DNS root service have taken on even greater importance. Any lapse or decline in root service performance could potentially trigger widespread network connectivity issues. Traditional monitoring methods often yield limited insights and fail to accurately reflect the true status of the root service [1, 2]. To address this challenge, this paper presents a monitoring system designed specifically for the DNS root service status. This system leverages a combination of active and passive monitoring techniques to provide real-time oversight of the root server's operational status, thereby offering robust and dependable support for domain name resolution. Active monitoring mimics user requests, periodically dispatching queries to the root server to assess latency and response rates. Complementing this, passive monitoring involves analyzing recursive resolution logs from servers within a designated network. This approach extracts root request and response data, shedding light on key metrics such as the network's recursive resolution status, root service request traffic, and its distribution.

2 System Design and Implementation

2.1 System Architecture and Interface

The system is composed of three layers: data acquisition, analysis, and presentation. The data acquisition layer handles the execution of active monitoring tasks and gathers passive monitoring data. Subsequently, the analysis layer takes charge of purifying and analyzing the amassed data, formulating monitoring indicators, and conducting anomaly detection for the root service. Lastly, the presentation layer renders these indicators visible via a graphical interface, enabling an intuitive comprehension of the root service's status. Figure 1 showcases the monitoring interface specifically tailored for visualizing the root service status of a provincial government's network domain name system, operating within a live production environment.



Fig. 1. Monitoring interface of the system

2.2 System Implementation

2.2.1 Active Monitoring

Active monitoring uses the DNS protocol to send query requests to root servers, recording response times and resolution outcomes. By leveraging multi-threading technology, we can perform concurrent queries, significantly boosting monitoring efficiency. Furthermore, we utilize scheduled tasks to ensure regular execution of active monitoring. The primary monitoring indicators we focus on are as follows:

Root Server Latency: Employing a distributed testing methodology, we conduct tests on all configured root servers for the recursive server every minute. This allows us to

calculate the average response time across all root servers, providing a clear picture of their performance.

Root Server Response Rate: Again, utilizing the distributed testing approach, all root servers set up for the recursive server undergo testing every minute. This enables us to determine the average response rate of the root servers, which is a crucial indicator of their reliability and responsiveness.

2.2.2 Passive Monitoring

Passive monitoring involves mirroring the DNS resolution traffic of the recursive server to a designated traffic analysis server. Utilizing high-performance traffic parsing and restoration techniques, crucial attribute information, including response time and response codes, is extracted and securely stored in a database for in-depth analysis. The key monitoring indicators that we focus on are:

The proportion of root server accesses, which reveals the distribution of visits across different root servers, providing insights into traffic patterns and potential bottlenecks.

Requested root server traffic, indicating the total volume of requests received per second by all root servers. This metric provides us with a comprehensive understanding of the request traffic from recursive servers to root servers.

Recursive request traffic and recursive response traffic, signifying the number of incoming requests and outgoing responses per second handled by all recursive servers. These figures illustrate the efficiency and responsiveness of the recursive servers.

Recursive resolution delay and recursive resolution success rate, reflecting the average time taken for DNS resolutions and the percentage of successful resolutions across all recursive servers. These indicators are essential for assessing the performance and reliability of the DNS system.

2.2.3 Root Service Anomaly Detection Based on Unsupervised Learning

An efficient and real-time anomaly detection method for the DNS root service has been devised, leveraging Apache Kafka and Flink. Kafka guarantees real-time data transmittance and efficient storage, whereas Flink offers robust real-time data processing and analytical prowess. This architecture not only fulfills real-time prerequisites but also exhibits remarkable scalability and versatility, facilitating seamless integration with various algorithms and services.

(1) Data Acquisition and Transmission

As a distributed stream processing platform, Kafka adeptly manages substantial volumes of real-time data streams. Both active and passive monitoring data from the DNS root service can be instantaneously relayed to the Kafka cluster via Kafka Producers. Kafka Topics serve to differentiate between distinct data stream types. For instance, separate topics can categorize active and passive monitoring data.

(2) Real-time Data Processing and Analysis

Flink, a high-performance, multipurpose big data processing engine, is suited for both batch and stream processing. Flink Consumers read real-time data streams from

Kafka, enabling real-time data processing, encompassing operations like data cleansing, transformation, and aggregation. Through Flink's window operations (e.g., tumbling windows or sliding windows), real-time statistics and analysis of DNS root service data are feasible, including computations for average response times and response rates.

(3) Anomaly Detection

Anomaly detection algorithms, such as Isolation Forest or other machine learning techniques, can be seamlessly integrated into the Flink processing pipeline. These algorithms perform anomaly detection on the real-time data stream, promptly identifying and flagging anomalous data points. When an anomaly is detected, an alert system can be activated to notify operational staff for prompt action.

(4) Result Output and Storage

The processed outcomes can be redirected back into designated Kafka topics for utilization by other systems or services. Additionally, these results can be archived in a database or data warehouse for subsequent offline analysis or report generation.

(5) Analysis of Anomaly Detection Results

Based on anomaly scores, potential outliers are pinpointed and scrutinized. Some conceivable anomalies and their possible causes encompass:

Abnormally High Query Frequency: A sudden escalation in queries targeting a specific TLD might suggest a DNS amplification attack on that TLD.

Unusual Query Time Distribution: A surge of queries during atypical hours (e.g., late evenings) could indicate automated scanning or nefarious activities.

Abnormal Query Sources: A notable volume of queries stemming from a singular IP address or subnet may signify targeted scanning or malevolent behavior.

Unusual Response Times: Considerably extended response times might denote an attack on the root domain name server or network congestion.

Through these methodical steps, we can effectively harness unsupervised anomaly detection methods to discern irregularities in DNS root resolution data. This ensures the stability and security of the network environment through prompt alerts and subsequent investigations by operational personnel.

3 Conclusion

This paper presents the design and implementation of a status monitoring system for the domain name system root service, leveraging both active and passive monitoring techniques. By integrating the strengths of active monitoring, passive monitoring, and anomaly detection, our system offers a thorough and precise representation of the root service's status. Extended production environment results demonstrate the system's high practicality and reliability, bolstering the stable operation of the root service.

Acknowledgments. This work was partially supported by National Key Research and Development Program of China (2024YFB31NL00105), the Major Key Project of PCL (PCL2023A05-005).

References

1. Wang, Q., Luo, M., Yao, Y., Xin, L., Jiang, Z., Shi, W.: Measurement for encrypted open resolvers: applications and security. *Comput. Netw.* **213**, 109081 (2022)
2. Yang, D., Li, Z., Jiang, H., Tyson, G., Li, H., Xie, G.: A deep dive into DNS behavior and query failures. *Comput. Netw.* **214**, 109131 (2022)



Dynamic Route Planning System Integrated with Traffic Flow Sensing

Bingkun Wang, Yixin Tian, Fangshu Chen^(✉), Jiahui Wang, and Yufei Zhang

School of Computer and Information Engineering, Shanghai Polytechnic University,
Shanghai 201209, China
fschen@sspu.edu.cn

Abstract. Route planning plays an important yet challenging role in global positioning system (GPS). Traditional static route planning methods are limited to static traffic conditions, ignoring the dynamic changes in traffic flow when urban traffic jams and accidents occur. Therefore, the dynamic route planning system named DRPT is proposed, which analyzes and predicts future road congestion based on real-time and historical traffic information and gives optimal routes under dynamic route planning. DRPT systematically consolidates map matching, road speed prediction, dynamic route planning, and road state visualization, fully considering the dynamic characteristics and spatio-temporal dependency of road network traffic status.

Keywords: Intelligent transportation system · Dynamic route planning · Traffic forecasting

1 Introduction

Route planning is a core component in GPS navigation and taxi applications, playing a crucial role [1]. However, deriving the optimal route is an NP-hard problem [2]. Traditional route planning algorithms can not satisfy the non-stationary dynamic traffic conditions and fail to meet users' demands for optimal routes [3, 4]. Furthermore, designs aiming for the shortest travel time in congested road conditions are more reasonable [5].

Based on the above problems, we propose a Dynamic Route Planning system under dynamic Traffic (DRPT) to cope with drastic changes in traffic conditions effectively and capture the temporal and spatial dependencies of roads. DRPT constantly updates network information and congestion levels, utilizing the updated traffic flow for dynamic route planning to identify the optimal routes. In terms of optimization, DRPT utilizes the spatio-temporal attention mechanism to capture the characteristics of nearby road segments and adopts an incremental approach to dynamically plan the travel time of a route.

B. Wang and Y. Tian—Contribute equally to this work.

The key contribution of our system is concluded as follows:

1. DRPT systematically integrates map matching, road speed prediction, dynamic route planning, and road condition visualization.
2. DRPT can dynamically plan an updated transportation road network and select the best route based on real-time traffic conditions.
3. DRPT displays predicted road speed information, dynamic route planning results, and roadway congestion heat maps to provide high-quality visualizations.

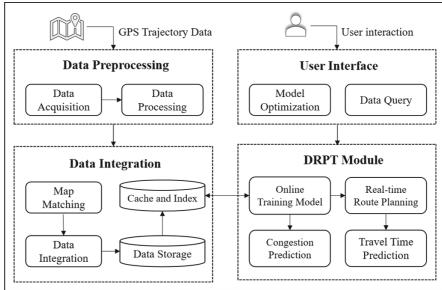


Fig. 1. System Architecture Diagram.

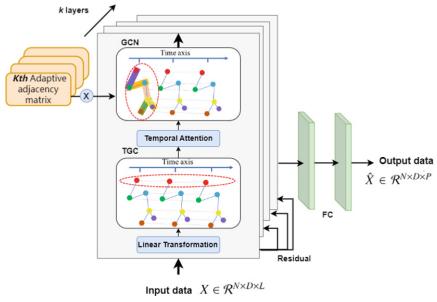


Fig. 2. The architecture of DiffSTG.

2 System Architecture

Figure 1 depicts the DRPT structure with the following four main modules:

Data Preprocessing. The module mainly collects and normalizes the original GPS track data. We eliminate outliers with zero displacements between adjacent points and speeds exceeding the road speed limit.

Data Integration. The module maps the processed GPS track data to the road network through the angle-based map-matching algorithm [6] and stores the mapped road network information in a database. The road network and associated track data are stratified and indexed by G-tree, and the algorithm searches the data module based on the current query point and reads it into the Redis cache, thereby improving the query efficiency.

DRPT Module. The module handles requests from front desk users and can predict travel time, traffic congestion and real-time route planning according to user needs. In Fig. 2, the DRPT module adopts our previous work using Diffusion Spatial and Temporal Graph convolution network (DiffSTG) to predict traffic flow [6]. Users can also train and optimize the model online through the interactive interface to obtain the optimal route and traveling time after training.

User Interface. This module is user-oriented and provides users with some interactive functions. Users send requests to the algorithm module for the result and a visual view.

3 Dynamic Route Planning

Traditional routing methods, such as the Dijkstra algorithm, typically plan routes based on network information available at the time of departure and are incapable of continuously providing the optimal routes in response to real-time changes in traffic flow and road conditions. To address these issues, we propose a Single-step Progressive Dijkstra (SPD) algorithm for dynamic route planning. Specifically, compared with the traditional static Dijkstra algorithm, the SPD algorithm has been improved in the following aspects:

Progressive Strategy. The SPD algorithm uses a progressive strategy to manage changes in the road network. This allows the algorithm to adapt and update route planning outcomes as conditions change.

Dynamic Adaptability. The SPD algorithm has dynamic adaptability, allowing it to adjust routes based on a weighted road network that changes over time to address traffic flow variations and congestion, thereby providing the optimal path.

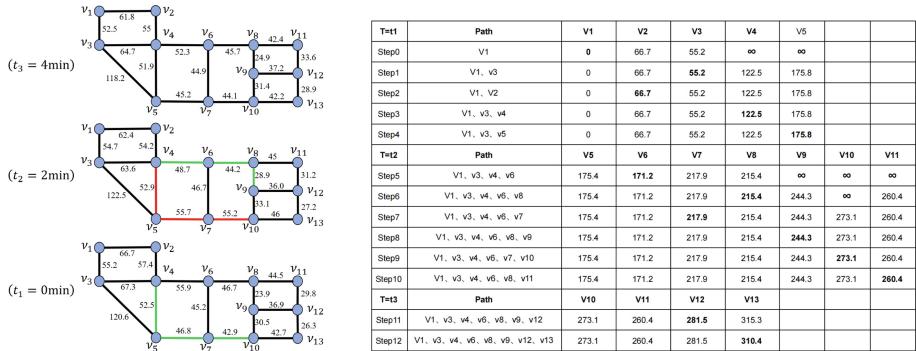


Fig. 3. An example to illustrate the algorithm mechanism of SPD.

As shown in Fig. 3, the route starts at v_1 and ends at v_{13} , and the SPD algorithm updates the network weight every two minutes. The shortest route obtained by Dijkstra at t_1 is (v_1, v_3, v_4) , which takes 122.5 s. At t_2 , the SPD algorithm detected that road congestion occurred on the red segment (v_4, v_5, v_7, v_{10}) and replanned the route to the green segment, specifically (v_4, v_6, v_8, v_9) , which takes 121.8 s to pass. At t_3 , the route is updated to (v_9, v_{12}, v_{13}) , and the passage time is 66.1 s. Therefore, the final route of the SPD algorithm is $(v_1, v_3, v_4, v_6, v_8, v_9, v_{12}, v_{13})$, and the total passage time is 310.4 s. The route obtained by Dijkstra's algorithm is $(v_1, v_3, v_4, v_5, v_7, v_{10}, v_{13})$, and the total passage time is 328.5 s.

From the above examples, we can see that the SPD algorithm can capture the dynamic changes in traffic conditions. Dynamic routes planned by the SPD

algorithm can adjust the route according to changes in real-time traffic conditions and provide real-time route planning results to select the optimal route. However, compared with other dynamic route planning algorithms, the SPD algorithm is a little bit time-consuming since the traffic forecasting block is updated every two minutes to capture the real-time traffic condition.

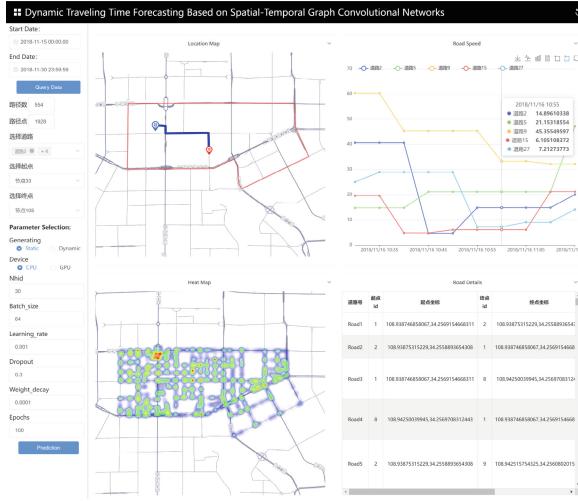


Fig. 4. User Interface Diagram.



Fig. 5. Route Planning.

4 System Demonstration

The system collected a total of 2 million track data from the didi platform from November 15, 2018, to November 30, 2018. When users interact with DRPT, they can choose the start point and the endpoint, and the system will draw the corresponding optimal route. Here are a few examples of interaction scenarios:

Scenario 1: DRPT User Interface. Fig. 4 displays the homepage of DRPT, where users can specify the following parameters: (1) Select a date range to view data; (2) Customize the selection of multiple roads for visualization of road speed in table or line chart form; (3) Specify starting and ending points for optimal route planning; (4) Users adjust the model by customizing parameters. By specifying different parameters, users view various visualizations in the process of interacting with the system, including road speed visualizations, road congestion visualizations, road detail displays, etc.

Scenario 2: Dynamic Route Planning. Users can specify the starting and ending points for dynamic route planning, and the system predicts road network traffic in real time, updating road weights every two minutes. Additionally, users

can customize parameters such as epochs, batch size, learning rate, and dropout, as shown on the left sidebar in Fig. 4, to train the model and plan the optimal route again. In Fig. 5, the top side shows static route planning with a travel time of 303.7 s. The bottom side shows real-time dynamic route planning with a travel time of 294.5 s. In addition, we randomly set 1000 source pairs and destination pairs, and the experiment demonstrates that the average execution time of the SPD algorithm is milliseconds and the average travel time of the SPD algorithm is 6.12% less than that of the Dijkstra algorithm.

References

1. Hu, D., Chen, L., Fang, H., Fang, Z., Li, T., Gao, Y.: Spatio-temporal trajectory similarity measures: a comprehensive survey and quantitative study. *IEEE Trans. Knowl. Data Eng.* **36**, 2191–2212 (2023)
2. Liu, H., Jin, C., Zhou, A.: Popular route planning with travel cost estimation from trajectories. *Front. Comp. Sci.* **14**, 191–207 (2020)
3. Dai, G., Hu, X., Ge, Y., Ning, Z.: Attention based simplified deep residual network for citywide crowd flows prediction. *Front. Comp. Sci.* **15**, 1–12 (2021)
4. Gao, Y., Fang, Z., Xu, J., Gong, S., Shen, C., Chen, L.: An efficient and distributed framework for real-time trajectory stream clustering. *IEEE Trans. Knowl. Data Eng.* **36**, 1857–1873 (2023)
5. Chen, F., Qi, Y., Wang, J., Chen, L., Zhang, Y., Shi, L.: Temporal metrics based aggregated graph convolution network for traffic forecasting. *Neurocomputing* **556**, 126662 (2023)
6. Chen, F., Zhang, Y., Chen, L., Meng, X., Qi, Y., Wang, J.: Dynamic traveling time forecasting based on spatial-temporal graph convolutional networks. *Front. Comp. Sci.* **17**(6), 176615 (2023)



NLITS: A Natural Language Interface for Time Series Databases

Yuting Lin¹, Jianqiu Xu^{1(✉)}, Xieyang Wang¹, and Yitong Zhang²

¹ Nanjing University of Aeronautics and Astronautics, Nanjing, China
{linyuting,jianqiu,xieyang}@nuaa.edu.cn

² Newcastle University, Newcastle upon Tyne, UK
c3061157@newcastle.ac.uk

Abstract. Time series data has become an important part of many fields. The management of time series data is becoming increasingly important due to the growing demand for applications. However, most users lack the ability to use specialized time series databases. Thus, we develop a natural language interface for time series databases called NLITS. The system comprises three components: data preprocessing, natural language understanding, and executable database statement generation. We preprocess natural language queries to extract and normalize time information. To extract the key entity information, we establish a time series data knowledge base and propose a context-based time series data parsing algorithm. Meanwhile, we train a model for query category recognition through a designed time series data corpus by using the LSTM network. Finally, NLITS generates executable database statements. Our demonstration will showcase how NLITS enables users directly querying time series databases with natural language. Experiments show that NLITS has a good performance with an average response time of 1.14s, a translatability of 91.7% and a translation accuracy of 88.9%.

Keywords: Time Series Databases · Natural Language Interface · Data Preprocessing · Semantic Parsing · Time Series Data Corpus

1 Introduction

Time series data is defined as an ordered collection of observations or sequence of data points made through time at often uniform time intervals [1]. With the rapid development of modern information technology, time series data has become an essential part of many fields, such as medical diagnosis, speech processing, financial analysis, and environmental monitoring. The storage, retrieval and management of time series data is becoming increasingly important due to the growing demand for applications [4]. However, non-technical users lack the ability to use professional time series databases, and structured query language is costly for non-technical users to learn. Therefore, the NLIDB (Natural Language Interface to Database) systems oriented towards time series databases still hold significant research value.

The study of NLIDBs could be traced back to the 1960s. The common NLIDBs are mainly based on relational databases, such as ATHENA++ [6] and IRNet [2]. Concurrently, quite a lot of research exists on NLIDBs for other types of databases, such as SpatialNLI [5] for spatial databases and NALMO [8] for moving objects databases. However, these NLIDBs have limitations in handling time series data. For example, NALMO parses time limited to simple expressions (e.g., 8 am and 10 am). Time expressions in time series data queries are plentiful and variable. As shown in Fig. 1, “In May, 2009” is not a specific time point, but an interval of time. The existing NLIDBs cannot adequately support time series data queries. Recently, the LLMs (Large Language Model) have rapidly emerged. While LLMs demonstrate proficiency in generating syntactically valid SQL statements, LLMs often struggle to produce semantically accurate queries [7].

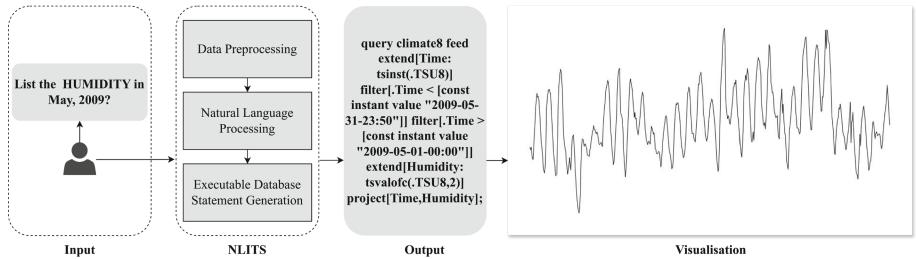


Fig. 1. The flowchart for processing a natural language query.

In this demonstration, we introduce a tool called NLITS (Natural Language Interface for Time Series Database), which is designed to help users to convert natural language queries into executable database statements. NLITS addresses the diversity of time expressions in terms of time granularity and time types, as illustrated in Fig. 1, and supports users to perform time point queries, time interval queries and aggregation queries in time series databases. In addition, the open-source extensible database SECONDO [3] is expanded to create a time series database that can be used to support time series data types and operations. Experiments showcases that NLITS provides efficient, accurate, and convenient methods for manipulating time series databases to help non-technical users obtain valuable information from time series data.

2 The Framework

Figure 2 gives the system overview and pipeline of NLITS. The system can be divided into NLP (Natural Language Processing) model layer and data layer. The NLP model layer provides the core technologies for NLITS, which can be classified into three phases: (i) *data preprocessing*, (ii) *natural language understanding*, and (iii) *executable database statement generation*.

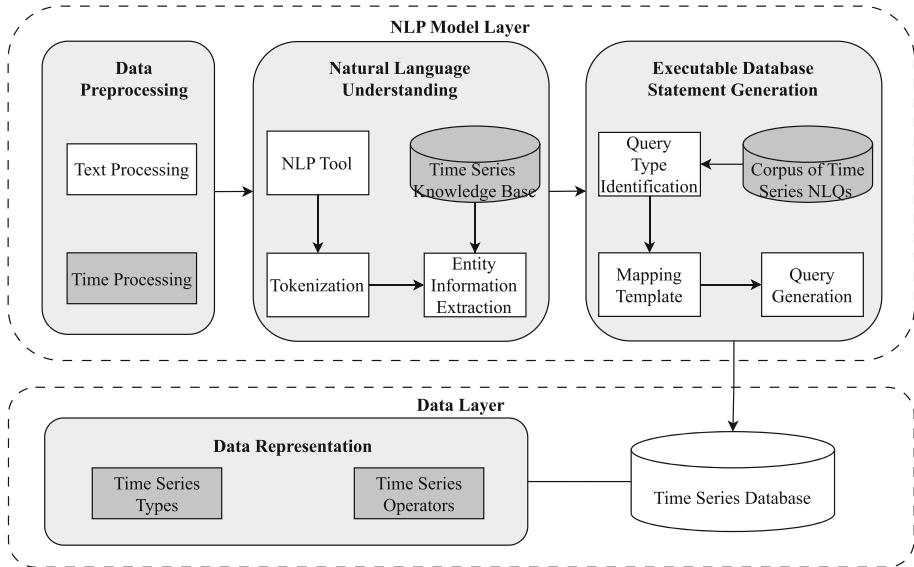


Fig. 2. The architecture of NLITS.

Task (i) is mainly to extract and normalize time expressions. Task (ii) is to extract the exact entities. Task (iii) is to validate the query type and generate executable database statements for manipulating time series databases. Meanwhile, the data layer includes the database and data representation for time series. In particular, data presentation indicates time series data types and operators for storing, retrieving and analyzing data in the time series database.

Data Preprocessing. We provide a time extraction algorithm to preprocess natural language queries, which is used to extract and normalize time expressions as the standard 24-hour format. The algorithm not only solves the diversity of time expressions, but also lays the foundation for entity extraction.

Natural Language Understanding. To extract the key entities, we utilize spaCy, a NLP tool, for tokenization and initial entity extraction. We build a knowledge base of time series data and extract accurate entities using a context-based time series data parsing algorithm. The algorithm improves the semantic parsing accuracy.

Executable Database Statement Generation. We build a time series corpus, and the accuracy of the query category recognition model is 99.75% by training the time series data corpus through the LSTM (Long Short-Term Memory) network. Finally, executable database statements are generated by entity mapping according to query type, which are used to access time series databases.

3 The Demonstration

The framework NLITS is implemented in an extensible database system SEC-ONDO. We prepare two time series datasets and store them as covid19 and climate8, as shown in the Table 1.

Table 1. Time series database statistics

Dataset	#Tuples	#Variables	Size
Coronavirus (Covid-19) Data	267010	5	11 MB
Climate Data Time-Series	420551	9	41.1 MB

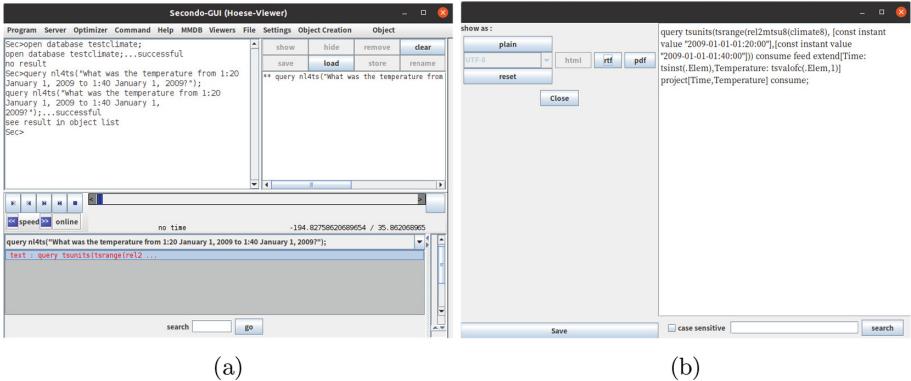


Fig. 3. Demonstration of NLITS.

Evaluation Indicators. These are three evaluation indicators: (i) *response time*, (ii) *translatability*, and (iii) *translation accuracy*. Response time is the total time spent on the natural language translation process. Translatability is the ratio of the number of natural language queries successfully translated into executable database statements to the total number of natural language queries. Meanwhile, translation accuracy is the ratio of the number of natural language queries that can be translated and successfully executed to the total number of natural language queries.

Test Cases¹. Based on covid19 and climate8, we construct 36 test cases including time point queries, time interval queries and aggregation queries to satisfy the diversity of natural language expressions as much as possible.

Demonstration Scenario. Figure 3 illustrates a time interval query. The user inputs “*What is the temperature from 1:20 January 2009 to 1:40 January 2009?*”, displayed in Fig. 3a. The system generates an executable database statement as shown in Fig. 3b.

¹ <https://github.com/RachelALin/NLITS/blob/main/Testcase.txt>.

Performance Evaluation. In the test cases, the longest response time of NLITS is 1.21 s, and the shortest response time is 1.06 s. The average response time, translatability and translation accuracy of NLITS are 1.14 s, 91.7% and 88.9% respectively.

Acknowledgments. This work was supported by the NSFC under Grant No.U23A20296 and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX24_0608).

References

1. Fu, T.: A review on time series data mining. *Eng. Appl. Artif. Intell.* **24**(1), 164–181 (2011)
2. Guo, J., Zhan, Z., Gao, Y., et al: Towards complex text-to-sql in cross-domain database with intermediate representation, pp. 4524–4535. Association for Computational Linguistics (2019)
3. Güting, R.H., Behr, T., Düntgen, C.: SECONDO: a platform for moving objects database research and for publishing and integrating research implementations. *IEEE Data Eng. Bull.* **33**(2), 56–63 (2010)
4. Ha, M., Shichkina, Y.A.: Translating a distributed relational database to a document database. *Data Sci. Eng.* **7**(2), 136–155 (2022)
5. Li, J., Wang, W., Ku, W., et al: Spatialnli: a spatial domain natural language interface to databases using spatial comprehension, pp. 339–348. ACM (2019)
6. Sen, J., Lei, C., Quamar, A., et al.: ATHENA++: natural language querying for complex nested SQL queries. *Proc. VLDB Endow.* **13**(11), 2747–2759 (2020)
7. Sun, S., Zhang, Y., Yan, J., et al.: Battle of the large language models: dolly vs llama vs vicuna vs guanaco vs bard vs chatgpt - a text-to-sql parsing comparison, pp. 11225–11238. Association for Computational Linguistics (2023)
8. Wang, X., Liu, M., Xu, J., et al.: NALMO: transforming queries in natural language for moving objects databases. *GeoInformatica* **27**(3), 427–460 (2023)



FOICP-Miner: An Interactive Spatial Pattern Recommendation System Based on Fuzzy-Ontology

Zhiwei Chen, Zezheng Geng, and Xuguang Bao^(✉)

Guilin University of Electronic Technology, Guilin 541004, China
baoxuguang@guet.edu.cn

Abstract. Spatial pattern mining is essential for the exploratory analysis of spatial data, with numerous efficient systems available that can discover various types of spatial patterns in large datasets. However, identifying patterns that are genuinely interesting to a specific user remains a significant challenge. To address this issue, we develop FOICP-Miner, an interactive system based on the fuzzy ontology. It is designed to facilitate the effective discovery of personalized spatial patterns tailored to individual user interests. FOICP-Miner employs the domain-specific fuzzy ontology to encapsulate users' background knowledge. Users are then prompted to express their preferences on sample patterns. Finally, the system leverages a fuzzy ontology model to evaluate the users' prior knowledge and extract their preferred patterns from the candidate pattern set. Our evaluation results demonstrate that FOICP-Miner identifies user-preferred patterns more effectively compared to the state-of-the-art researches.

Keywords: Spatial Pattern Mining · Interactive Process · Fuzzy Ontology · Do-main Knowledge

1 Introduction

The explosive growth of spatial data results in a significant demand for spatial data mining. Consequently, spatial co-location pattern mining, an important branch of spatial data mining, garners increasing attention in recent years. A spatial co-location pattern refers to a subset of spatial features that are frequently located in close proximity to each other. Typically, mining spatial co-location patterns involves using a user-specified minimum prevalence threshold to identify prevalent patterns. To avoid missing interesting spatial co-location patterns, very low prevalence thresholds are often set. This results in a large number of prevalent patterns, of which only a small proportion is user-preferred. Bao and Wang [1] proposed an approach using the ontology to estimate the semantic similarity between two co-location patterns, aiming to identify user-preferred patterns by explicitly constructing precise background knowledge. However, this method suffers from information loss due to the rigid boundaries of crisp relationships, leading to imprecise semantic similarity between patterns. Accordingly, using conventional ontologies reduces the accuracy of recommendations.

Ontologies allow domain knowledge to be captured in an explicit and formal manner, facilitating sharing between humans and computer systems. In conventional ontologies, there is a binary relationship between concepts, where 1 and 0 represent the existence or absence of a relationship. This leads to all child-parent relationships being equally weighted. However, concepts may not have semantically equal relationships with their parents. Fuzzy ontologies address this issue by handling fuzzy knowledge [2], where concepts are related with a degree of membership u ($0 \leq u \leq 1$). By allowing partial membership of one item to another, fuzzy ontologies capture richer semantics than traditional domain knowledge representations.

In this work, we propose FOICP-Miner, a system designed to interactively discover user-preferred patterns using the fuzzy ontology. Rather than returning all patterns for feedback and minimizing iteration rounds, we introduce an efficient sampling method for co-location pattern selection and an updating method for candidate patterns to alleviate users' burden.

2 System Overview

FOICP-Miner undergoes seven steps to interactively help users effectively discover preferred co-location patterns according to their specific interests, as shown in Fig. 1.

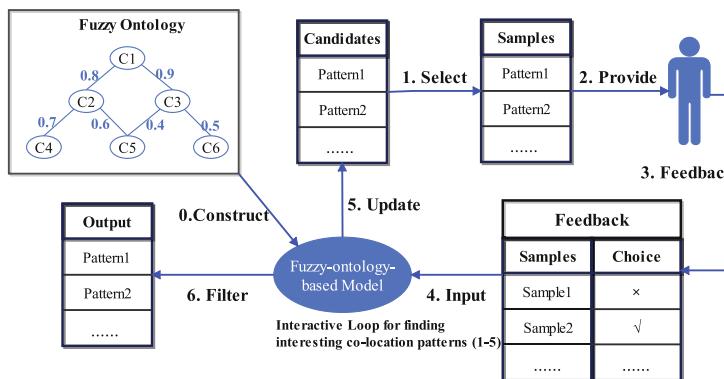


Fig. 1. Framework description of FOICP-Miner

Step 0: Fuzzy Ontology Construction: The fuzzy ontology can be created by domain experts or reused from existing ones, as there is an increasing number of ontologies available on the Semantic Web.

Step 1: Sample Pattern Selection: The system selects several sample patterns for the user. The number of samples can be decided by the user. The system adopts a greedy selection strategy to choose these sample co-location patterns [3].

Step 2: Displaying Samples: The selected sample co-location patterns are displayed to the user.

Step 3: User Feedback: The user provides feedback by indicating whether they like the co-location pattern or not.

Step 4: Incorporating Feedback: FOICP-Miner takes the user's feedback and candidates as input to the fuzzy-ontology-based model.

Step 5: Similarity Estimation and Filtering: The fuzzy-ontology-based model uses the fuzzy Jaccard similarity measure [4] to estimate the semantic similarity between co-location patterns. It first calculates the similarities of each sample co-location pattern with every candidate. Then, it moves selected samples and their similar patterns to the output, while removing uninteresting samples from the candidates. If the candidate set is not empty, FOICP-Miner returns to Step 1 to continue the interaction.

Step 6: Output Interesting Patterns: After the interactive process, all interesting co-location patterns are stored in the output.

3 Demonstration Scenario and Efficiency Evaluations

FOICP-Miner is designed with a user-friendly interface, making it simple for users to interact with the system. To demonstrate FOICP-Miner, we used a subset of the points of interest (POI) data from Beijing [3]. The dataset includes 20 features and 28,900 instances. The prevalence threshold is set as low as 0.1, because we aim to identify some rare co-location patterns. The source code and detailed references are available on GitHub¹, while the video demonstration is also accessible on YouTube².

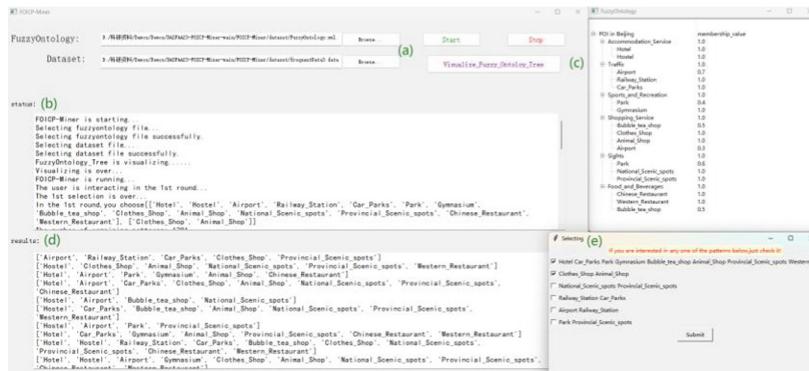


Fig. 2. Demonstration of FOICP-Miner

Demonstration. Figure 2 shows the main interface of FOICP-Miner. In Fig. 2a, an XML file with the description of a fuzzy ontology and a file with prevalent co-location patterns are required. Figure 2b shows the runtime information including the current

¹ The code of this paper can be found in <https://github.com/czuest/APWeb-WAIM-FOICP-Miner>.

² The video demonstration can be viewed on <https://youtu.be/Di8X3TnwAjw>.

operation, the patterns that the user selected in each round, and the number of current candidates. The visualize button shows the visualization of the fuzzy ontology as a tree structure in Fig. 2c. Figure 2d shows the personalized interesting co-location patterns for the user. Figure 2e shows the interaction interface which requires the user to select his/her interesting sample co-location patterns.

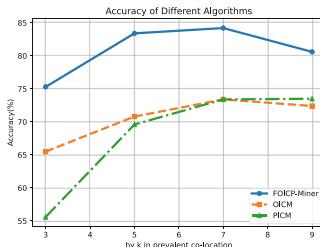


Fig. 3. Efficiency comparison with related literature

To evaluate the performance of FOICP-Miner, we simulated user preferences and performed multiple interactions with the system. We compared the accuracy [1] of FOICP-Miner with OICM [1] and PICM [5] by varying the number of sample co-locations k within 6 rounds. Results indicate that with the gradual increase of k , accuracy first rises and then drops. This is because a smaller k does not provide sufficient information for effective updates, while a larger k increases the likelihood of conflicting choices. In conclusion, the system efficiently recommends reliable, interesting patterns to users, saving significant manpower and resources (Fig. 3).

4 Conclusion

Existing spatial co-location mining methods often generate many prevalent co-location patterns, but only a small proportion of these are of real interest to users. In this demonstration, we develop a system using the fuzzy ontology to interactively find interesting patterns, helping users effectively discover prevalent patterns that match their specific interests. The demonstration scenarios show that FOICP-Miner can further assist users in making better decisions and enhance the reliability of their decision-making.

Acknowledgments. This study was funded by National College Students' innovation and entrepreneurship training program (202310595020) and Innovation Project of Guangxi Graduate Education (2023YCXS041).

References

1. Bao, X., Wang, L., Chen, H.: Ontology-based interactive post-mining of interesting co-location patterns. In: Asian Pacific Web Conference, pp. 406–409 (2016)

2. Huitzil, I., Bobillo, F.: Fuzzy ontology datatype learning using Datil. *Expert Syst. Appl.* (2023)
3. Bao, X., Gu, T., Chang, L., et al.: Knowledge-based interactive postmining of user-preferred co-location patterns using ontologies. *IEEE Trans. Cybern.* (2021)
4. Cross, V., Mokrenko, V.: Fuzzy set similarity between fuzzy words. *IFSA/NAFIPS* (2019)
5. Wang, L., Bao, X., Cao, L.: Interactive probabilistic post-mining of user-preferred spatial co-location patterns. In: 2018 IEEE 34th International Conference on Data Engineering (2018)



SPCCP-Miner: Towards the Discovery of Congested Junctions

Zheyng Liu, Zhengyu Yang, and Xuguang Bao^(✉)

Guilin University of Electronic Technology, Guilin 541004, China
baoxuguang@guet.edu.cn

Abstract. Traffic congestion occurs when the total traffic volume on the road network exceeds the road's capacity, disrupting normal traffic flow and causing psychological stress for city residents. Junctions as the focal points of traffic volume in urban road networks, play a crucial role in alleviating traffic congestion through their rational planning. Thus, to assist traffic management departments in alleviating traffic congestion, we present a system called SPCCP-Miner (Sub-Prevalent Co-Location Congestion Pattern Miner) to discover congested junctions. The system identifies congested intersections based on spatial-temporal data mining by discovering sub-prevalent patterns. In addition, by utilizing density peak clustering algorithm, the system's efficiency has been effectively enhanced. Upon analyzing the congested junctions discovered by the system, users can make decisions to remit traffic congestion, such as extending the green time. Experimental evaluations using real-world datasets validate the system's efficacy.

Keywords: spatio-temporal data mining · traffic congestion · congested junction

1 Introduction

As urbanization advances and private car ownership increases, urban traffic congestion has become increasingly severe. It is often observed that when there is congestion on segment A of a road, its adjacent segment B also experiences congestion simultaneously. The primary cause of this phenomenon is the congestion at the junction between segments A and B. Therefore, the strategic identification and efficient management of congested traffic intersections are paramount for mitigating urban traffic congestion.

Considering the problem of traffic congestion, previous studies have primarily focused on congested roads and explored the causes of road congestion from the perspective of the roads themselves, often neglecting the influence of junctions on traffic congestion [1]. To solve the above problems, we present a novel and efficient system, named SPCCP-Miner (sub-prevalent co-location congestion pattern miner), to discover congested junctions.

In SPCCP-Miner, the spatial-temporal interconnections among congested roads at varying times can be derived using the spatio-temporal mining algorithm. Considering the proximity of roads at intersections, we employ the sub-prevalent pattern mining method to uncover congested junctions. To enhance system efficiency, we utilize the

density peak clustering algorithm for a parallel clustering process. Based on the identified congested intersections, users can formulate strategies to alleviate traffic congestion, significantly contributing to the mitigation of urban traffic congestion.

2 System Overview

The architecture of our proposed system is shown in Fig. 1. The core of the system is divided into three layers.

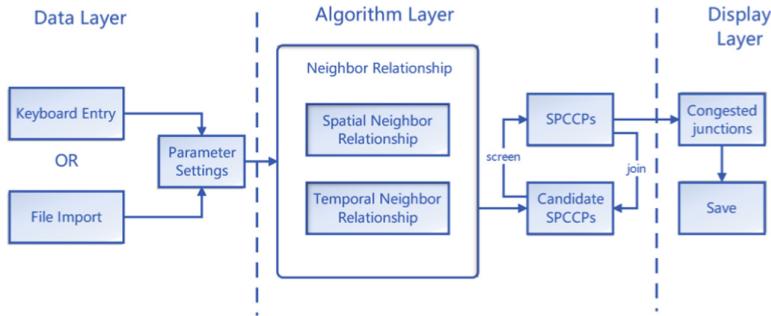


Fig. 1. The framework of the SPCCP-Miner

Data Layer. We provide users with two methods to upload their dataset: keyboard entry and file import.

Algorithm Layer. The neighbor relationship between congested roads at different times can be generated via spatio-temporal mining algorithm [1]. Distinct from other spatio-temporal co-location fuzzy congestion pattern mining methods [1–3], SPCCP-Miner employs an efficient density peak clustering algorithm [4] and sub-prevalent pattern mining method [5] to achieve higher completeness in identifying congested junctions. We improve the algorithm efficiency by parallelizing the clustering process.

Display Layer. After generating all SPCCPs, the results of the algorithm are displayed on the interface of SPCCP-Miner.

3 Demonstration Scenarios

SPCCP-Miner features a user-friendly interface, offering a straightforward interaction for users. In this demonstration, we utilize a subset of a real-world traffic dataset from Guiyang City [3] to illustrate the capability and efficiency of SPCCP-Miner. The source code¹ and video demonstration² of this system are accessible via the provided links.

¹ <https://github.com/soulying/SPCCP-Miner.git>.

² <https://youtu.be/U3AQKxxcQ7A>.

Demonstration. Figure 2 presents the main interface of SPCCP-Miner. The uploaded data is shown in Fig. 2a, parameters are listed in Fig. 2b, and all generated SPCCPs are displayed in Fig. 2c. For example, Fig. 3 shows various urban roads labeled with unique capital letters. By inputting congestion data and road neighbor relationships, the system reveals two size-2 SPCCPs, indicating frequent congestion at the junctions from road A to road C, and road G to road J.

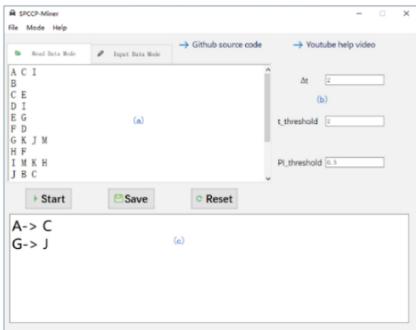


Fig. 2. Demonstration of SPCCP-Miner

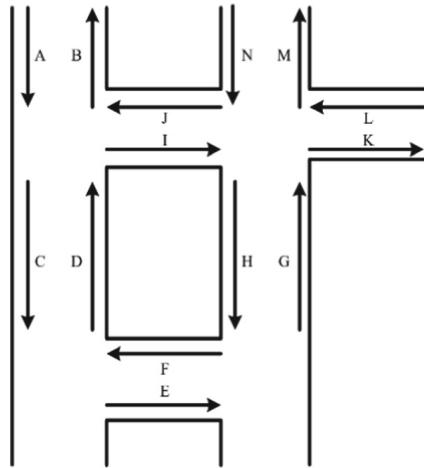
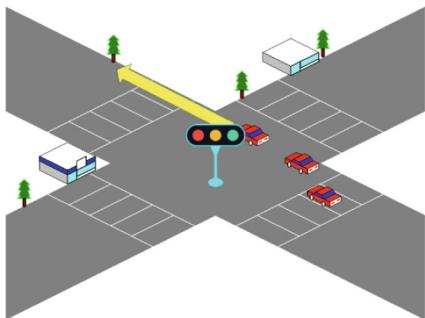
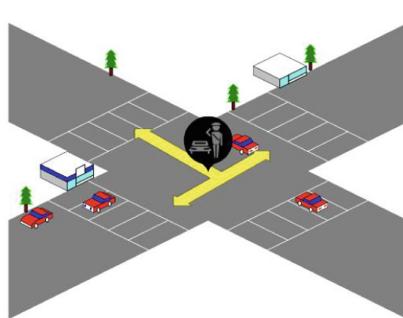


Fig. 3. Example of road segments

Analysis of Results. We can further analyze SPCCPs to propose additional recommendations for mitigating traffic congestion. For example, if the size of a SPCCP is 2, as shown in Fig. 4a, congestion can be alleviated by extending the green light duration. For size-3 or larger junctions, as depicted in Fig. 4b, deploying additional traffic officers can help mitigate congestion.



(a) The solution to size-2 SPCCP



(b) The solution to size-3 SPCCP

Fig. 4. Different solutions for different sizes of SPCCP

Efficiency Evaluations. We evaluated the efficiency of SPCCP-Miner in two aspects. First, compared to TCPMS-FCP [3], our algorithm significantly reduces runtime by parallelizing the clustering process. Second, we assessed the effects of different PI thresholds on junction congestion determination. As shown in Fig. 5, increasing the PIthreshold significantly reduces the running time for generating SPCCPs. This is because higher PI-thresholds eliminate many large SPCCPs, and once the PI-threshold reaches a certain level, only small SPCCPs remain.

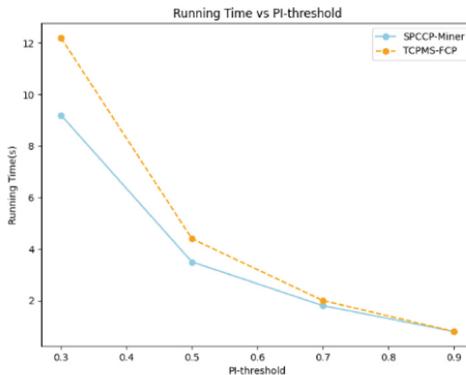


Fig. 5. PI-threshold impacts on running time

4 Conclusion

In most cases, congested junctions are significant causes of traffic congestion. Distinct from other approaches in this field which mainly concentrate on congested roads themselves, this demonstration presents a novel system called SPCCP-Miner to discover congested junctions. By analyzing congested junctions, traffic management departments can propose solutions to alleviate traffic congestion.

Acknowledgments. This study was funded by National College Students' innovation and entrepreneurship training program (S202410595290) and Innovation Project of Guangxi Graduate Education (2023YCXS041).

References

1. Wang, X., Wang, L., Wang, J.: Mining spatio-temporal co-location fuzzy congestion patterns from traffic datasets. *J. Tsinghua Univ. (Sci. Technol.)* **60**(8), 683–692 (2020)
2. Yang, L., Wang, L.: Mining traffic congestion propagation patterns based on spatiotemporal co-location patterns. *Evol. Intell.* **13**(2), 221–233 (2020)

3. Wang, X., Wang, J., Wang, L., Wang, S., Ding, L.: TCPMS-FCP: a traffic congestion pattern mining system based on spatio-temporal fuzzy co-location patterns. In: Chbeir, R., Huang, H., Silvestri, F., Manolopoulos, Y., Zhang, Y. (eds.) *Web Information Systems Engineering – WISE 2022*. WISE 2022. LNCS, vol. 13724. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20891-1_47
4. Zhou, T., et al.: Fuzzy regional co-location pattern mining based on efficient density peak clustering and maximal fuzzy grid cliques. *J. Data Sci. Intell. Syst.* (2024)
5. Wang, L., Bao, X., Zhou, L., Chen, H.: Mining maximal sub-prevalent co-location patterns. *World Wide Web* **22**(5), 1971–1997 (2019)



Crowd-OBIGA: A Crowdsourced Approach for Oracle Bone Inscriptions Glyph Annotation

Zhaoan Dong^{1,2(✉)}, Xiaofan Wang¹, Jing Xiong^{1,3}, Guangshun Li^{1,2}, and Qingju Jiao³

¹ School of Computer Science, Qufu Normal University, Rizhao, China
dzan@qfnu.edu.cn

² Rizhao-Qufu Normal University Joint Technology Transfer Center, Rizhao, China

³ Key Laboratory of Oracle Bone Inscriptions Information Processing, Ministry of Education, Anyang, China

Abstract. High-quality glyph key point annotation data plays a crucial role in the study of Oracle Bone Inscriptions (OBI) detection and recognition. However, the presence of noise, such as vague scratches and incomplete or irrelevant natural marks on most oracle bone fragments significantly impacts the accuracy of machine algorithms in glyph recognition. Although OBI experts can rely on their professional experience to avoid the impact of noise, the number of experts is small and costly especially for large-scale labeling needs. Crowdsourcing, as a prevalent and cost-effective solution for data annotation leveraging human visual recognition abilities via sophisticated task design, reduces the difficulty and cost of oracle bone glyph annotation. In this paper, we introduce a crowdsourcing tool for annotating glyph key points in OBI. Through a simple interactive interface, crowdsourcing workers can provide key point annotation data, thereby offering abundant training data for machine learning-based oracle bone glyph recognition and classification.

Keywords: Oracle Bone Inscriptions · Glyph annotation · Crowdsourcing

1 Introduction

Oracle Bone Inscriptions (OBI) represent the earliest form of systematic Chinese characters discovered [1]. Recently, there has been a growing interest from both institutions and individuals in the study of OBI [2]. With the advancement of digital technology, research on OBI has increasingly shifted from traditional manual analysis to automatic recognition and classification using machine learning techniques. Existing research materials on OBI includes photographs, rubbings, and facsimiles, with a primary focus on glyph key points. These glyph key points are derived from the extensive expertise of OBI experts and represent abstract forms of individual OBI characters, independent of specific material carriers. The OBI characters on the inscribed pieces vary in size, orientation, and distribution, significantly increasing the complexity of detection and recognition [3].

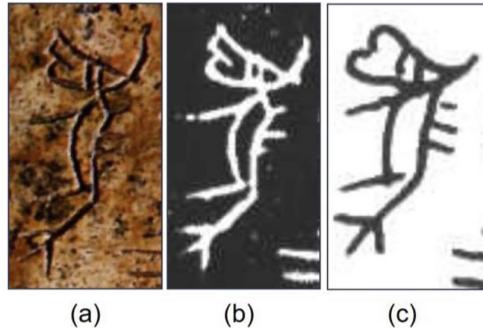


Fig. 1. Different forms of “馬(horse)”: (a) photograph, (b) rubbing, and (c) facsimile

As shown in Fig. 1, the character “馬(horse)” exhibits issues such as incompleteness, wear, and scratches on the OBI fragments, leading to subtle variations in its forms across different representations and reducing machine recognition accuracy [4].

Crowdsourcing, as a powerful paradigm for problem-solving especially for tasks that are difficult for computer to resolve solely [5, 6], can effectively leverage human visual recognition abilities to mitigate various noise interferences, and through sophisticated task design, reduce the difficulty and cost of oracle bone glyph annotation. Crowdsourcing worker can readily identify and annotate noise, such as scratches on OBI. This system envisions workers adjusting and refining key points on another annotated font image, guided by the glyph keywords predefined by experts, thereby enhancing the system’s functionality.

The workflow of Crowd-OBIGA is as follows: Once registered, crowdsourcing workers must complete task tutorials and pass a qualification test before accepting tasks. Workers can then modify key points on a glyph image previously annotated by an automatic algorithm, using an expert-defined glyph as a reference. After submitting their edits, the system saves the results in JSON format and awards points based on the completion and accuracy of their tasks.

2 System Design and Implementation

2.1 System Architecture

The architecture of Crowd-OBIGA is illustrated in Fig. 2. System architecture. It comprises three layers: the user interface layer, the task logic layer, and the storage layer.

2.2 User Interface Layer

System users are divided into Requesters and Workers. Users register and log in according to their roles to access different user interfaces and complete different task requirements. Overall, it means that the requester uploads the task to the system based on their own needs, and the worker selects the tasks to be completed and submitted.

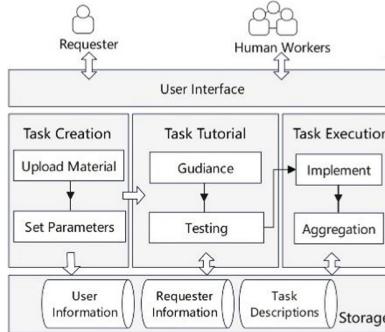


Fig. 2. System architecture

2.3 Task Logic Layer

This part primarily presents the system flowchart, which is divided into three main modules: Task Creation, Task Tutorial, and Task Execution.

Firstly, the task creation module is responsible for initializing the task and setting parameters. Secondly, the system provides task guidance to explain the task requirements and operation steps. The guidance module include task descriptions, operation manuals. Thirdly, a testing phase is conducted to verify whether the workers have the capability to execute them. If the workers pass the test, they move on to the task execution module. During the task execution process, workers need to follow the task requirements and steps to complete specific operations. Finally, after the task is completed, the system aggregates the results and returns them to the requester.

2.4 Storage Layer

The system employs MySQL for database, including requester information, user information and task descriptions. The requester and user information are stored in text format, while the task description is saved in JSON format, including content such as character ID, the number of key points, and the coordinate location information of each key point. The original image data used for task description in the system comes from the HWOBC dataset [7], containing 83,245 character-level samples which are grouped into 3881-character categories.

3 Demonstration Scenarios

We intend to showcase the Crowd-OBIGA system through the following scenarios:

Task Creation: The requester user interface encompasses two primary functions: the task creation interface and the task viewing interface. These interfaces empower the requester to effortlessly create and manage various tasks, ensuring effective execution and monitoring.

Task Execution: After logging in, the user is guided to complete the test. For instance, as shown in Fig. 3. Task Page, the worker adjusts the yellow movable dots on the annotation image according to the green dots displayed in the reference image. This includes tasks such as moving and adding dots. Once the test is finished, the system evaluates whether the worker has passed. Only upon passing can the worker proceed with the subsequent tasks.

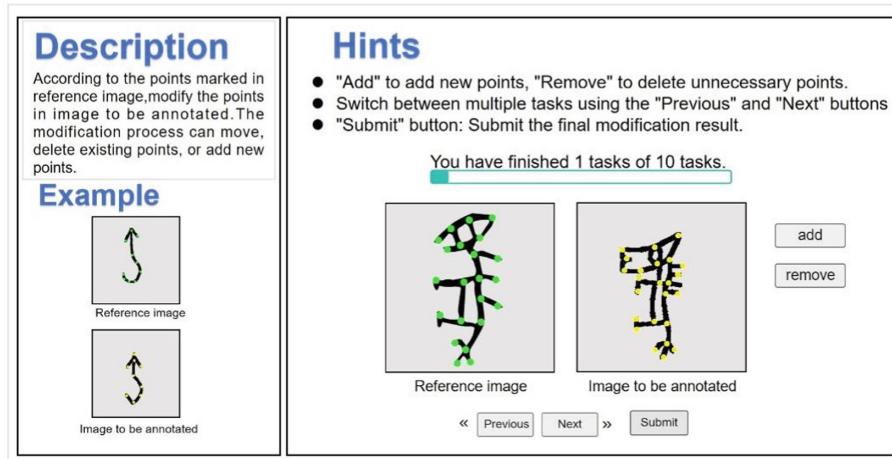


Fig. 3. Task Page

Acknowledgments. This work is supported by Shandong Provincial Natural Science Foundation (No.ZR2020MF149), the Key Technology Project of Henan Educational Department of China (22ZX010), the National Natural Science Foundation of China (U1504612), the Chinese Ministry of Education and National Language Commission Special Project for Research and Application of Oracle Bone Inscriptions and other ancient characters (YWZ-J023).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Xiong, J., Liu, G., Liu, Y., Liu, M.: Oracle bone inscriptions information processing based on multi-modal knowledge graph. *Comput. Electric. Eng.* **92** (2021). <https://doi.org/10.1016/j.compeleceng.2021.107173>
2. Xiong, J., Guo, T., Liu, J., Chen, Y.: OBSKP: oracle bone studies knowledge pyramid model with applications. *Knowl* **49**(7), 483–495 (2022). <https://doi.org/10.5771/0943-7444-2022-7-483>
3. Liu, G.: Oracle bone text detection and recognition based on deep learning. *Yindu J.* **41**(03), 54–59 (2020). <https://doi.org/10.16140/j.cnki.ydjk.2020.03.011>

4. Yinqi Wenyuan. <http://jgw.aynu.edu.cn/>. Accessed 5 June 2024
5. Dong, Z., Jiaheng, L., Ling, T.W., Fan, J., Chen, Y.: Using hybrid algorithmic-crowdsourcing methods for academic knowledge acquisition. *Cluster Comput.* **20**(4), 3629–3641 (2017). <https://doi.org/10.1007/s10586-017-1089-8>
6. Dong, Z., Lu, J., Ling, T.: PANDA: a platform for academic knowledge discovery and acquisition. *BigComp* 10–17 (2016). <https://doi.org/10.1109/BIGCOMP.2016.742579>
7. Li, B., Dai, Q., Gao, F., Zhu, W., Liu, Y.: HWOBC-A handwriting oracle bone character recognition database. *J. Phys: Conf. Ser.* **1651**(1), 012050 (2020). <https://doi.org/10.1088/1742-6596>



NexusDB: A Large-Scale Distributed Time-Series Database for Industrial Scenarios

Linlin Ding, Di Yuan Chzhen, Yuda Li, Zhiyong Zhang, Zhiran Xie,
and Mo Li^(✉)

School of Information, Liaoning University, Shenyang, China
{dinglinlin,limo}@lnu.edu.cn

Abstract. With the development of Industrial Internet of Things (IIoT) and sensor technology, the speed and volume of data generation in industrial environments are rapidly increasing. In these settings, sensors continuously generate large volumes of data, traditional databases often fall short due to their focus on generality rather than continuous data flow. We propose NexusDB, a distributed time-series database tailored for such environments. NexusDB optimizes storage structures and query execution, offering streamlined architecture, high performance, minimal footprint, and robust distribution capabilities. Experimental results show that NexusDB significantly outperforms conventional time-series databases in both data insertion and querying, particularly in large-scale scenarios.

1 Introduction

As the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) advance, they generate an immense volume of time-series data within the industrial sector [3]. For example, microseismic monitoring systems utilize gravity and charge sensors to continuously produce significant amounts of data. The efficient storage and management of this data are critical for effective real-time analysis and prediction. Distinct from traffic or social network data, industrial big data exhibits unique characteristics such as fixed frequency, device offline scenarios, real-time requirements, and low data redundancy [4].

Typical challenges of industrial time-series data are shown in Fig. 1, including missing data points, delay data points or sensor offline. Existing systems, have been developed to manage time-series data, yet they struggle to effectively address the unique characteristics of such data in the industrial domain. QuestDB [5], with its column-oriented design, boasts remarkable single-node write performance but lacks in distributed deployment capabilities and scalability. IoTDB [7], designed for IoT scenarios, utilizes LSM for data persistence but shows inferior performance when dealing with degraded data quality conditions.

It can be observed that above industrial characteristics require a different approach to handling industrial time-series data compared to generic time-series

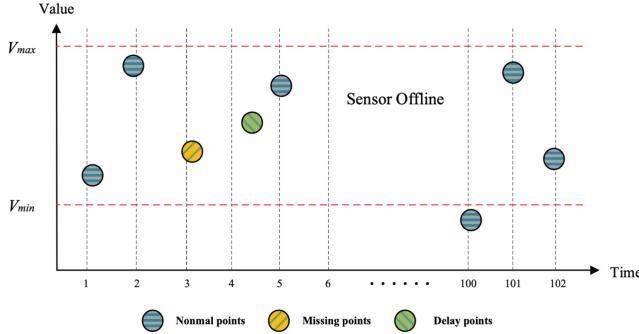


Fig. 1. Typical Challenges of Industrial Time-Series Data

data (e.g., financial data, where the timing of orders cannot be assumed and precise timestamps need to be recorded). NexusDB introduces a suite of innovative features through a variety of methods, chief among them being: (i) eschewing the direct storage of timestamps in favor of (ii) utilizing memory-based index offsets to denote time differences, (iii) bypassing the LSM approach for data persistence while distinctly separating indexing from execution processes, and (iv) implementing the parallelization of multiple execution workers to enhance efficiency. Therefore, the distinctive advantages of NexusDB can be summarized as follows: i) **Extremely Fast**, ii) **Lightweight**, iii) **Scalable**, iv) **Simple interfaces**. The experimental results of NexusDB on large datasets show a significant improvement compared to other mainstream databases. This also confirms that in specific domains, specialized solutions outperform general-purpose solutions. Currently, NexusDB, as a nascent database, has been deployed in several industrial projects. Its modular design abstraction allows for easy integration of various optimization strategies, leaving ample room for further enhancements in the future (Fig. 2).

2 System Design

System Architecture: NexusDB simplifies the database by removing various features that are unnecessary for industrial scenarios, based on the aforementioned characteristics. Its core storage structure is a data format, which can also be abstracted as a file [1], referred to as NexusFile (NF) hereafter. NF is a pure data segment without containing metadata such as data tags. Built upon NF is an Index Manager that indexes the physical locations of NFs. All operations (writes, queries) need to be converted through the Manager to obtain the physical location of NFs, and the Manager also controls operations such as NF creation, merging, and deletion.

NexusFile Storage Structure. For most TSDBs, timestamp columns typically use 64-bit storage, which is costly even when various compression schemes

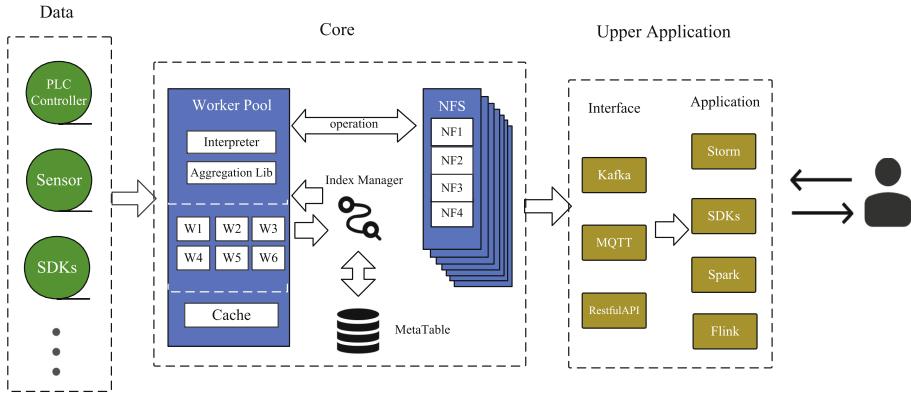


Fig. 2. NexusDB Architecture Overview

are employed. Essentially, the storage cost of timestamp information cannot be avoided [1]. For sequences with relatively fixed time intervals, the timestamp can be represented using indices in memory plus starting timestamp. For values with some deviation from standard time intervals, they can be rounded to the nearest slot. Admittedly, this sacrifices a certain degree of accuracy in the original data, but for industrial scenarios, there is effectively no difference between data at 37.024s and data at 37s when conducting various calculations (Fig. 3).

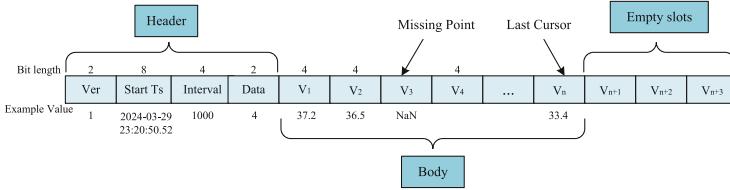


Fig. 3. NexusFile Bit Layout

The beginning of NF consists of a fixed-length data header, comprising only 16 bits: 2 bits for the version number, 8 bits for the start timestamp, 4 bits for the time interval, and 2 bits for the data length. The subsequent storage size is $4 \times n$, where n is the amount of data.

Index Manager. The NF alone lacks self-consistency and cannot even determine which data it describes or what type of value it represents. Therefore, the use of NF relies on management by the Manager (Fig. 4).

The Manager organizes all time series in a hierarchical tree structure, where each leaf node corresponds to a time series. This approach shares similarities

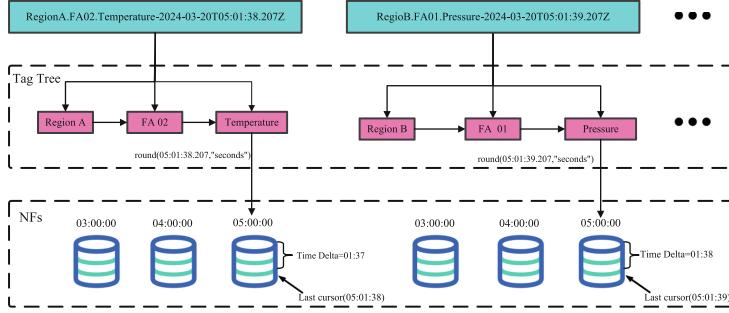


Fig. 4. Index Manager Search Example

with IoTDB [8] in terms of trade-offs, but we also transform the hierarchical structure into tags, similar to assigning a unique ID to each sensor.

NexusDB treats multi fields sensor as separate sensors, while tags are limited to strings separated by “.”. When a request arrives, the Manager uses tag decomposition to quickly locate the position of the NFs for the data series and determines the operation to be performed based on the specific timestamp t .

3 Evaluation

We selected three databases for comparison: InfluxDB [2], TimescaleDB [6], and QuestDB [5]. We first assess the write throughput, which represents the rate of data points ingested into the database per second, for each database across various time-series scales. The results have been shown in Fig. 5(a). Remarkably, the query efficiency of NexusDB is clearly better than all the competitors in all tests, as illustrated in Fig. 5(b).

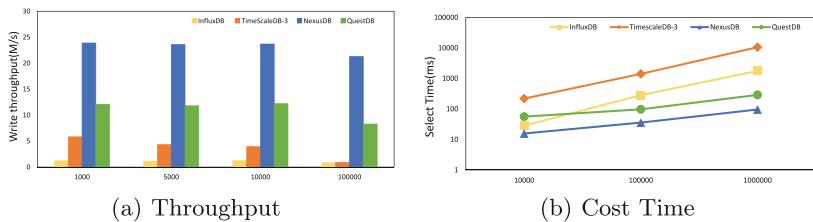


Fig. 5. Evaluation

Acknowledgements. This study was funded by the National Natural Science Foundation of China (No.62072220); Natural Science Foundation of Liaoning Province (2022-KF-13-06, 2022-BS-111); Liaoning Provincial Department of Education Youth Project (No. JYTQN2023189).

References

1. Andersen, M.P., Culler, D.E.: Btrdb: optimizing storage system design for timeseries processing. In: FAST, pp. 39–52 (2016)
2. Inc., I.: Influxdb (2024). <https://docs.influxdata.com/influxdb/v2.6/>. Accessed 07 Apr 2024
3. Liu, P., Guo, X., Shi, B., Wang, R., Wo, T., Liu, X.: Non-salient region erasure for time series augmentation. *Front. Comput. Sci.* **16**(6), 166349 (2022). <https://doi.org/10.1007/S11704-022-1765-6>
4. Liu, S., et al.: QAAS: quick accurate auto-scaling for streaming processing. *Front. Comput. Sci.* **18**(1), 181201 (2024)
5. QuestDB: Questdb (2024). <https://questdb.io/>. Accessed 07 Apr 2024
6. TimeScale Inc.: Timescaledb (2024). <https://www.timescale.com>. Accessed 07 Apr 2024
7. Wang, C., Huang, X., Qiao, J., et al.: Apache iotdb: time-series database for internet of things. *VLDB* **13**(12), 2901–2904 (2020)
8. Wang, C., Qiao, J., Huang, X., et al.: Apache iotdb: a time series database for IoT applications. *Proc. ACM Manag. Data* **1**(2), 195:1–195:27 (2023)

Industry Paper



LMStor: Storage Acceleration Design for Large Models

Biyun Shang¹, Feng Zhang², Mo Xu^{2(✉)}, Junning Xu², and Zhenjiang Dong¹

¹ Nanjing University of Posts and Telecommunications, Nanjing, China

² Nanjing Zhongxing New Software Company Limited, Nanjing, China

xu.mo1@zte.com.cn

Abstract. Recently, large-scale and diverse training datasets have become the key to superior semantic understanding capabilities of large models. As the number of parameters grows, large models have increasingly higher requirements for data volume during training, which poses challenges to traditional storage systems. This leads to problems such as high overhead for checkpoint fault tolerance, slow data loading that hampers computational efficiency, and significant consumption of storage resources. To address these issues, we propose a storage acceleration design for large models, LMStor, to improve training efficiency and storage space utilization of large models. Experimental results demonstrate that LMStor reduces checkpoint file saving time by 92.99%, increases the loading operations per second (OPS) of small files by 7.46 times, and shortens the data loading process by 86.6%. Furthermore, LMStor effectively conserves storage resources.

Keywords: Large Models · Model Training · Storage Systems · Distributed Clusters · Checkpoints strategy

1 Introduction

Large models represented by ChatGPT [3] with excellent performance in tasks such as text generation and semantic understanding have attracted widespread attention from industry and academia. Given the outstanding emergence ability and strong generalization capability demonstrated by hundreds of billions of parameter models, researchers have pursued the expansion of parameter scale for large models. From GPT1 to GPT4, model parameters increase from 0.1 billion to trillions, and the amount of training data also increases from 5GB to tens of trillions of tokens. As model parameters and training datasets expand, the cost of training large models also rises rapidly. Thus, training costs have become the determin obstacle restricting the further development of large models.

The model training process is intertwined with data calculation, reading, and writing. Due to the immense size of model parameters and intermediate data, storage for those data becomes a problem that can not be ignored. Based on storage performance, the storage acceleration technologies proposed in existing

work for model training can be summarized into three categories: 1) Distributed memory management technology based on the large model [9] aims to reduce the communication overhead caused by data transmission in distributed clusters, which relies on the distribution and transmission strategy of model data in the GPU cluster. 2) Large model training memory access-aware heterogeneous storage technology [1, 10] uses the memory access pattern during the model training process to design data prefetching and offloading strategy to cover the problems caused by data movement between various storage media. 3) Large model data reduction technology [4, 7] depends on the data characteristics of the model to reduce data precision, thereby reducing the storage space required for model training.

Although the mentioned research has been devoted to storage optimization during the training process from multiple aspects, these technologies all rely on model structure and training tasks and are problematic to implement independently in the storage system. Thus, this paper starts from the decoupling of storage optimization and computing framework in the model training process, analyzes the load characteristics and IO stack of the life-cycle of model training from the perspective of storage, and then proposes a storage acceleration design, LMStor, for large models to improve training efficiency and storage utilization. Specifically, LMStor is designed for checkpoint files, reading overhead, and storage utilization. The main contributions are as follows:

- 1) Proposing a two-stage asynchronous parameter checkpoint technology to reduce the calculation pause caused by saving large model checkpoint files.
- 2) Designing a dataset read-acceleration technology based on pre-reading and caching to improve the reading OPS of small files.
- 3) Introducing a hot&cold storage strategy and an adaptive conversion technology to efficiently utilize the required storage resources while ensuring performance.

2 Storage Challenges of Large Models

Based on the storage perspective, the paper starts by analyzing the IO characteristics and existing problems in the current storage system at each stage of the entire life-cycle of model training. As shown in Fig. 1, the large model training process can be roughly divided into three stages: data collection and processing, model training, and model inference. In the first stage, storage-related tasks are assignments for the massive collected data that needs to be stored and saved after data cleaning. As the model size enlarges, its demand for storage capacity further increases. The most immediate problem facing large-capacity data storage is how to maximize storage utilization without harming reliability and read-and-write performance. Next is the model training, which is strictly related to storage I/O. Finally, the model inference stage mainly involves calculating and outputting through the trained model, and the requirements for storage IO are relatively low.

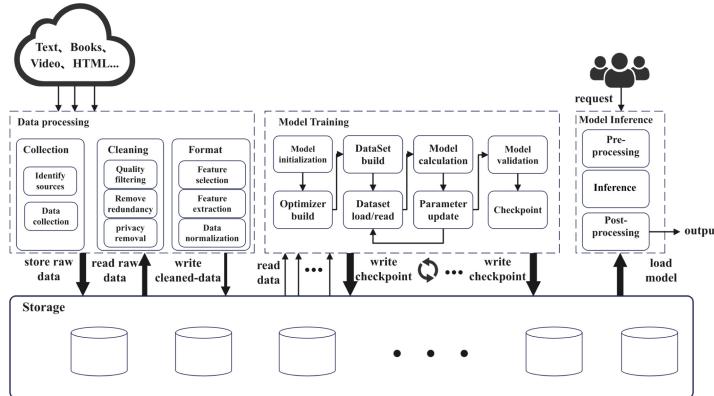


Fig. 1. Overview of large model training.

During the model training phase, we conduct an in-depth analysis of the IO stack and find out that excessive read-and-write waits leading to GPU idleness is one of the main reasons for low computing power utilization. As displayed in Fig. 2, we used an analysis tool to capture and analyze the IO stack, time occupancy, overhead, and read-and-write waits during the training of two typical workloads (large language model (LLM) and multimodal large model (MLLM)). Figure 2(a) offers the complete training process of a large language model. The processes ①~⑧ represent several important stages in the model training process. First, the model initialization and pre-training preparations are carried out, including model loading, tokenizer and optimizer initialization, and other pre-training works. Then the tokenizer is responsible for converting the dataset into individual tokens. Next, the two processes ③ and ④ represent building and loading the dataset from the underlying storage. The ⑤~⑥ are the training processes, in which model calculation and parameter updating are performed alternately. After a certain period of training is completed, the model verification process is performed to evaluate the accuracy of the model. Finally, the trained model parameters are written to the checkpoint file along with other information. From the time usage percentage, we can see that the total data loading time of the entire process accounts for 0.77%, while the saving time of a single checkpoint takes 583 s, and the total writing time of checkpoint files reaches 7.27%. Due to the waiting caused by read-and-write, the computing efficiency of the GPU is difficult to maximize.

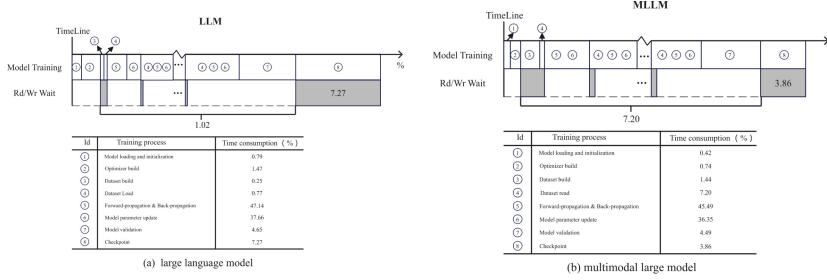


Fig. 2. Model training process and time-consuming proportion of LM training

Figure 2(b) pictures the training process of the multimodal large model and the time-consuming proportion of each stage. Compared with Fig. 2(a), the training stages are similar. With the difference in the training dataset (mostly picture files in multimodal large models), the total data loading time during the training process reaches 2933 s, and the total time cost accounts for 7.2%. The total computing pauses caused by reading and writing reaches 11.06%. Therefore, for datasets mainly composed of small files, an unavoidable problem in model training is how to reduce the time occupancy of periodic loading of training data.

In summary, there are three challenges for the storage system in the existing large model training: 1) Large checkpoint files interrupt model processing during saving checkpoint files. 2) Slow loading for small files procrastinates the training process. 3) Huge training data results in high costs for efficient storage.

3 Design And Implementation

In response to the above challenges, we propose the LMStor, designed to accelerate storage for large models. LMStor can effectively improve training efficiency and storage utilization without compromising the behavior of upper-level compute nodes. The elaboration of the three optimization techniques is as follows:

3.1 Two-Stage Asynchronous Parameter Checkpoint Technology

From the previous analysis, given the possible software and hardware failures encountered during the model training process, it is necessary to regularly save important parameters as checkpoints during model training to avoid re-training. For the checkpoint saving strategy, synchronous saving ensures data reliability strictly, but the intensive IO operations can lead to prolonged pauses in GPU computation. Asynchronous saving, on the other hand, writes to disk asynchronously without affecting foreground computations, thus significantly reducing computational pauses caused by IO. Although asynchronous saving can effectively resist process-level failures, it is still challenging to guarantee data reliability in the case of node failure. Hence, we combine the advantages of synchronous

and asynchronous to propose a highly reliable two-stage asynchronous parameter checkpoint technology. The strategy aims to preserve both data reliability and system efficiency while providing a more robust guarantee for the model training process.

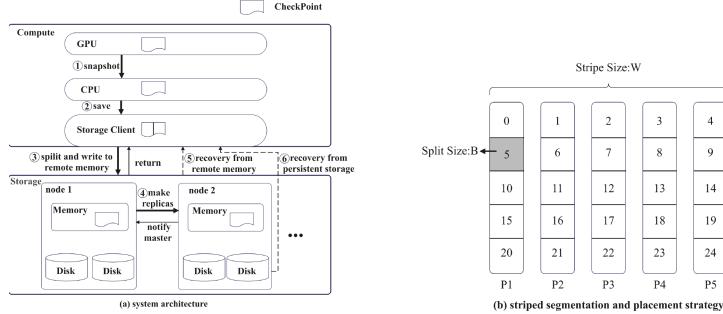


Fig. 3. Schematic diagram of two-stage asynchronous parameter checkpoint technology

As revealed in Fig. 3(a), the checkpoint writing process is classified into two stages. First, when the training process reaches the checkpoint saving stage, model data is copied from the GPU memory to the memory on the compute node. After the data is serialized, it is transferred to the storage node by calling the write interface of the storage client. The storage client then divides the data in the memory into blocks of a specific size and writes them to different storage nodes in parallel. When receiving the data block, the storage node saves it in memory and distributes a data copy to another node through the back-end high-performance network of the storage cluster to form dual replica storage. Once the storage node completes the replica construction in the memory, it notifies the client that the write operation is completed, and the upper-layer process will be resumed. Subsequently, the second phase focuses on checkpoint persistence. The background thread flushes the checkpoint files in memory to the disk. This asynchronous disk flushing can prevent disk IO from affecting compute nodes. The dual replica design ensures the data reliability of checkpoint files, reducing the risk of data loss from a single node failure.

For the purpose of maximizing the use of throughput of the storage node, shown in ③, data is divided into blocks for a single file. On the premise that network and disk bandwidth are not affected, block size should be set as small as possible so that the workload can be spread more evenly across more storage nodes and their disks. However, storing these small blocks in different files can lead to a metadata explosion that adversely affects performance and capacity. To this end, we design a strip file segmentation and placement strategy. As shown in Fig. 3(b), the file is split into blocks of size B (e.g., 128K) on the client, and then these blocks are sequentially placed into a strip of width W . When the number of data blocks exceeds the strip width, it is wrapped back to the starting position of the strip and continues to be placed. Each data placement position P in the

strip is mapped to a specific disk. This design not only significantly improves the parallelism of reading and writing files, but also effectively alleviates the problem of uneven load that may occur between disks. For each disk, different data blocks in the same file are aggregated and stored close to each other. When reading and writing files, data blocks are located by their offsets in the file.

In the two-stage asynchronous parameter checkpoint writing process, data is rapidly written to the memory of the storage node in phase one. Meanwhile, the memory of the storage node forms a distributed memory cache layer, which is related to the introduction of independent distribution. Compared with the introduction of an independent distributed memory cache layer, this method is simpler in structure and does not require any modification to the computing layer. In addition, memory management is performed by storage nodes rather than compute nodes, which helps the persistence layer implement a more efficient caching strategy. When a compute node fails, LMStor can quickly load the checkpoint file from the remote memory, thereby greatly reducing the fault recovery time. Only when the storage nodes fail, does LMStor need to load the checkpoint from the nearest checkpoint file.

3.2 Dataset Read Acceleration Technology Based On Pre-Reading And Caching

By analysis of the IO stack of multimodal large models, we observe that IO characteristics during the training process manifest as the loading of a large number of small files. Moreover, the data reading patterns during training are known in advance. Combining the above attributes in the training process, we propose a pre-reading and cache technology designed for model training.

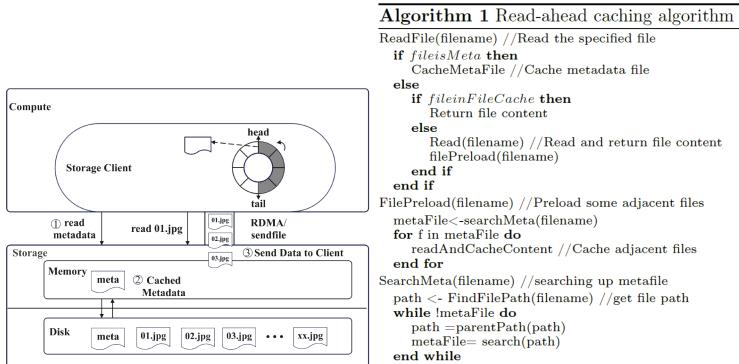


Fig. 4. Schematic diagram of storage-node read-ahead cache designed

The main architecture design is depicted in Fig. 4. When the compute node initiates a request to load a file to the storage node (as shown in ①), the storage node uses the metadata read-ahead cache algorithm (shown in Algorithm

1) to cache the metadata of the file(as shown in ②). Meanwhile, LMStor pre-reads more subsequent content according to a certain proportion based on the order in the metadata file and sends it to the computing node (as shown in ③). At the same time, to achieve faster data transmission, the storage client on the compute node requests a fixed size of memory when the process starts. The fixed-size memory requested by the compute node is structured as a ring buffer. Thus, the compute node directly loads data from memory. This read-ahead awareness strategy improves the efficiency of data loading and solves the problem that metadata cannot be effectively cached when loading a large number of small files.

3.3 Hot&cold Storage Strategy Adaptive Conversion Technology

As mentioned above, the current dataset used for training has exceeded one trillion tokens and is still growing, which has led to an increase in the size of checkpoint files. The management of these data has brought new challenges to the storage system. EC (Erasure Code) mode is a method of fault tolerance for data through encoding technology, which has superior storage efficiency. However, considering strict consistency, the traditional EC mode will force the data to be persisted into the disk. Therefore, the EC performance is limited by the disk IO. Regarding the characteristics that training datasets and checkpoint files are not updated after being written, LMStor adopts the method of adaptive consistency EC. At the same time, for the consideration of performance and cost, LMStor adopts a hot and cold storage strategy capable of ensuring large-capacity storage and guaranteeing performance.

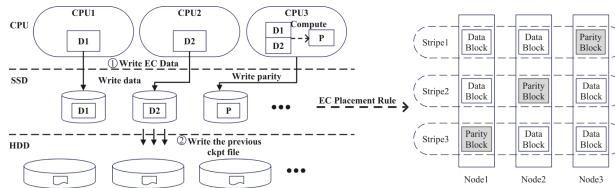


Fig. 5. Adaptive consistency EC strategy

When saving the checkpoint file, the replica placement strategy complies with defined rules (described in Sect. 3.1), that data replica in an EC strip is placed on the same host node. As demonstrated in Fig. 5, the strip first calculates the parity block in the memory and then persists it into different SSD disks (as shown in ①). In each strip, different physical blocks take turns to play the role of parity blocks, making the distributed storage system overall load balanced. Compared with the traditional method, adaptive consistency EC effectively avoids the disk bandwidth overhead caused by strict consistency. At the same time, in order to speed up model saving and recovery, LMStor uses a high-performance SSD as a

speed disk to speed up this process. When a new checkpoint is ready to be written to storage, the previous checkpoint file saved in the SSD is considered as the data that has become cold. Meanwhile, LMStor triggers background threads to migrate the old checkpoint file from SSD to HDD (as shown in ②) to achieve a balance between high performance and large storage capacity by masterly managing hot&cold data.

4 Experiments

4.1 Experiment Setup

Based on the designed experimental plan, we established a test cluster and conducted comparative tests before and after the optimization. The entire testing cluster consists of 22 servers: 16 compute nodes (each equipped with 8 Nvidia A100 GPUs) and 6 storage nodes (each with 512 GB of memory, 2 Intel P4610 NVMe SSDs, and 8 Western Digital HC550 HDDs). Clusters using the RoCE network for communication, with a bandwidth of up to 100Gb/s. Meanwhile, we select the Baichuan-13B [14] and Qwen-VL [2] large models as the experimental objects, both renowned for their exceptional performance.

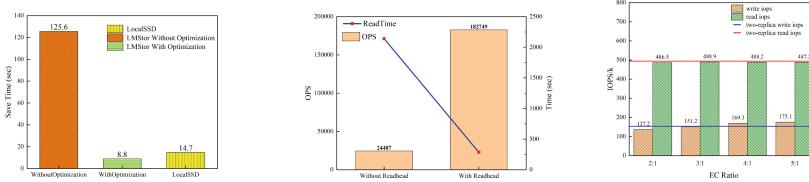


Fig. 6. Checkpoint saving time

Read-Ahead

Fig. 7. The influence of Read-Ahead

Time (sec)

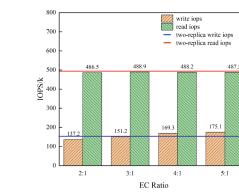
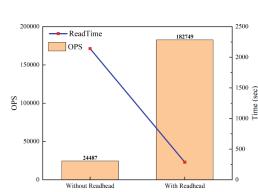


Fig. 8. Influence of adaptive EC strategy

4.2 Checkpoint File Saving Optimization

As shown in Fig. 6, the saving time of a single checkpoint file is reduced by 92.99% before and after the optimization. In addition, the saving time of LMStor after optimization decreases by 40.14% compared to local NVMe SSDs. This is mainly attributed to the high-bandwidth transmission between memory, which effectively shortens the saving time. At the same time, compared to using local NVMe SSD for write caching, the distributed LMStor storage system plays a critical role in resource aggregation and unified scheduling, which is more conducive to performance improvement.

4.3 Read-Ahead Optimization

Figure 7 suggests the influence on data loading during model training before and after adopting the read-ahead cache optimization strategy. It can be observed that after the optimization, the OPS (Operations Per Second) for small files

increases from 24,487 to 182,749, which is a 7.46-fold increase. Additionally, the total time required for data loading during the training process was reduced by 86.6%. The storage cache layer, designed based on the characteristics of data loading, effectively isolates the influence of the underlying storage on the training process, ensuring efficient data reading during training.

4.4 Adaptive EC Scheduling Strategy

Figure 8 portrays the IOPS for 64K sequential read-and-write under different ratios of EC stripes and a 2-replication mode. Under the adaptive consistent EC strategy, when the EC ratio is 5:1, the sequential write IOPS increases by 15.05% compared to the replica mode, while the read performances are nearly identical. The main reason is that after the EC is calculated in memory, less data needs to be written compared to a replica mode of the same scale. Therefore, the adaptive consistent EC strategy proposed in this paper, which is based on specific data storage characteristics, maintains efficient large-capacity storage without affecting the performance of data reading and writing.

4.5 Overall Model Performance Analysis

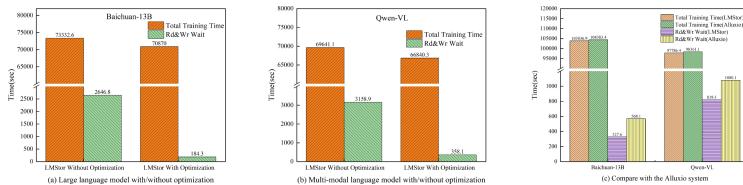


Fig. 9. Overall performance changes of LM before and after storage optimization

Figure 9 exhibits the overall training time and read-and-write waiting time of the large model before and after the LMStor optimization. In the training process of BaiChuan-13B, the read-and-write waiting time reduces from 2646.8 s to 184.3 s, a shortening of 93.04%, and the training time decreases by 3.36%. While in the training process of the Qwen-VL model, the calculation pause time caused by read-and-write waiting reduces by 88.66%, and the total training time reduces by 4.02%. In summary, LMStor reduces the time and resource consumption during the model training process. Figure 9(c) illustrates the comparative performance between LMStor and Alluxio under identical workload and hardware configurations (6 computing nodes and 6 storage nodes). Alluxio is a high-performance distributed caching system that leverages the memory on the computation side as a cache layer to accelerate the read-and-write during the training. As depicted in the graph, LMStor has reduced the read-and-write waiting time during the training of the BaiChuan and the Qwen-VL large model

by 42.3% and 24.1%, respectively, compared to Alluxio. The primary reason is that Alluxio places its cache layer on the computation side, which can consume bandwidth on the computation side during asynchronous saving. Additionally, Alluxio cannot utilize the cache for acceleration during the initial loading of the dataset. In contrast, LMStor is closely attuned to the dataset’s characteristics, and its asynchronous saving process occurs between storage clusters, resulting in a lesser impact on the training process.

5 Related Work

Checkpoint Technology: Most existing mainstream models use a synchronous method to save parameter checkpoint files during the model training, which strictly guarantees the reliability of the checkpoint while resulting in a severe waste of computing power. Wang [11], Mohan [8], Wu [12], and others have discussed the effect of asynchronous checkpoints in the training process. The asynchronous method can tremendously improve training efficiency, but their methods all rely on the training framework.

Data Loading Acceleration: The [5] training framework proposes a multi-stage parallel pipeline technology and efficient caching algorithm to accelerate the reading process of the dataset. In addition, there are some open-source deep learning libraries such as DALI [15], prefetch_generator [6], and DataLoader [13] accelerate the data loading process through multi-threading or asynchronous prefetching. However, those methods are changes made from intrusion into the computing layer and depend on the specific training framework implementation.

Large-Capacity Storage: Single-machine storage cannot guarantee the storage capability and fault tolerance for massive data. Therefore, the underlying storage for large model training usually uses distributed file systems such as HDFS. These storage systems generally can only support replicas or a certain proportion of the single EC storage strategy. From our analysis, the replica strategy will cause a waste of resources, while the traditional EC redundancy strategy will increase unnecessary disk IO and affect performance due to consistency guarantee.

6 Discussion

Existing checkpoint saving and dataset loading optimization techniques are usually closely coupled with the computation layer, relying on specific training frameworks or structures, making it difficult to flexibly adapt to the diverse needs of large model training. LMStor, based on the design concept of decoupling computation and storage, improves the read-and-write experience during the large model training process through storage-side optimization without intruding on the computation-side services and can adapt to any computation

framework. During the asynchronous checkpoint saving process, data needs to be transferred from the GPU to the host memory and then sent to the storage-side memory via the network for saving. Hence, the host-side memory, acting as a transfer, adds a redundant overhead to the entire saving process. Next, we will work on better coordination between asynchronous checkpoints and training frameworks, as well as optimization of system resource usage.

7 Conclusion

Based on the idea of decoupling storage optimization from the computing framework during large model training, we proposed LMStor, a storage acceleration design for large models, which optimized the data reading, writing, and storage behavior during large model training from the perspective of the storage system. Through the optimization of LMStor, while ensuring high reliability, the checkpoint file saving time is reduced by 92.99%, and the small file loading OPS is increased by 7.46 times. Compared with dual copies, the 5:1 EC redundancy mode increases storage utilization by 66.67%, effectively saving storage resources.

Acknowledgements. This work was supported by the National Key Research and Development Program of China (2021YFB3101101).

References

1. Bae, J., et al.: {FlashNeuron}:{SSD-Enabled} {Large-Batch} training of very deep neural networks. In: 19th USENIX Conference on File and Storage Technologies (FAST 2021), pp. 387–401 (2021)
2. Bai, J., et al.: Qwen-vl: a versatile vision-language model for understanding, localization, text reading, and beyond (2023)
3. Brown, T., et al.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
4. Chen, J., et al.: ACTNN: reducing training memory footprint via 2-bit activation compressed training. In: International Conference on Machine Learning, pp. 1803–1813. PMLR (2021)
5. Chen, L.: Deep Learning and Practice with Mindspore. Springer, Cham (2021). <https://doi.org/10.1007/978-981-16-2233-5>
6. Ishii, Y., Inaba, M., Hiraki, K.: Access map pattern matching for high performance data cache prefetch. *J. Inst.-Level Parall.* **13**(2011), 1–24 (2011)
7. Micikevicius, P., et al.: Mixed precision training. arXiv preprint [arXiv:1710.03740](https://arxiv.org/abs/1710.03740) (2017)
8. Mohan, J., Phanishayee, A., Chidambaram, V.: {CheckFreq}: frequent, {Fine-Grained} {DNN} checkpointing. In: 19th USENIX Conference on File and Storage Technologies (FAST 2021), pp. 203–216 (2021)
9. Narayanan, D., et al.: Pipedream: generalized pipeline parallelism for DNN training. In: Proceedings of the 27th ACM Symposium on Operating Systems Principles, pp. 1–15 (2019)

10. Rhu, M., Gimelshein, N., Clemons, J., Zulfiqar, A., Keckler, S.W.: VDNN: virtualized deep neural networks for scalable, memory-efficient neural network design. In: 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 1–13. IEEE (2016)
11. Wang, Z., et al.: Gemini: fast failure recovery in distributed training with in-memory checkpoints. In: Proceedings of the 29th Symposium on Operating Systems Principles, pp. 364–381 (2023)
12. Wu, B., et al.: Transom: an efficient fault-tolerant system for training LLMS. arXiv preprint [arXiv:2310.10046](https://arxiv.org/abs/2310.10046) (2023)
13. Xu, J., Wang, G., Yao, Y., Li, Z., Cao, C., Tong, H., et al.: A deep learning dataloader with shared data preparation. *Adv. Neural. Inf. Process. Syst.* **35**, 17146–17156 (2022)
14. Yang, A., et al.: Baichuan 2: open large-scale language models. arXiv preprint [arXiv:2309.10305](https://arxiv.org/abs/2309.10305) (2023)
15. Zolnouri, M., Li, X., Nia, V.P.: Importance of data loading pipeline in training deep neural networks. arXiv preprint [arXiv:2005.02130](https://arxiv.org/abs/2005.02130) (2020)



Enhancing Emergency Communications via UAV-Assisted Home-Independent Broadband Mobile Networks

Yiping Zhang^{ID} and Haobin Shi^{(✉) ID}

School of Computer Science, Northwestern Polytechnical University,
Xi'an 710072, China
{2021302607, shihaobin}@mail.nwpu.edu.cn

Abstract. Natural disasters often lead to significant communication breakdowns due to direct damage and indirect effects on communication infrastructure. Advances in unmanned aerial vehicle (UAV) technology, combined with its integration into wireless networks, present promising avenues for mitigating these challenges. However, a primary limitation of current solutions is that UAV-carried base stations are tied exclusively to a single telecommunications provider, limiting service access to subscribers of that provider and excluding others, particularly affecting user equipment (UE) in disaster zones. To overcome this barrier, we propose a UAV-assisted home-independent broadband mobile communication network that allows UEs from multiple operators to connect to a unified network and access broadband services. We have developed the architecture for this network and a signaling process that enables UEs to connect to this home-independent cellular network. This architecture allows UEs associated with various telecom operators to join the proposed network and establish data links efficiently. Extensive simulation results demonstrate the effectiveness of our proposed network architecture, showing good performance in signal coverage, packet transmission rate, round-trip time, and signal strength, which shows that our proposed scheme works very well.

Keywords: Emergency communications · unmanned aerial vehicle · multi-operator networks · home-independent communication

1 Introduction

Natural disasters often cause significant damage to the public switched telephone network and public land mobile network, leading to communications outages [1]. Rapidly restoring communication systems is crucial for aiding victims and coordinating rescue efforts by exchanging essential information, with emergency communication systems designed to function despite the challenges posed by natural disasters [2]. For traditional telecom networks, restoring communication often entails time-consuming and complex deployment of emergency stations or repair of local base stations [3, 4].

Recent advancements in unmanned aerial vehicle (UAV) technology and its integration with wireless networks like 4G and 5G, offer a transformative solution for emergency communications [5,6]. Within those systems terrestrial-satellite communication channels act the transmission links as substitutes of the damaged transmission networks on the ground [7,8].

In emergency rescue operations, tethered UAVs equipped with 4G/5G base stations and communication satellites as backhaul channels for core network access serve to act as substitutes for damaged ground communication stations [9]. Although this strategy mitigates some challenges, it also presents significant limitations. Specifically, the restoration of communication services in disaster areas is carried out independently by a single telecom operator. Only the user equipments(UEs) registered with this telecom operator can pass through the permission authentication of this telecom operator's network, therefore the UAV-carried base stations in disaster areas only provide communication services to users registered with this operator and do not extend benefits to users registered with other operators. This issue is crucial in disaster contexts, where spatial constraints hinder the deployment of multiple UAVs.

To address the gap in existing research on UAV-assisted emergency communication, we propose a home-independent broadband mobile communication network architecture in this paper. The multiple contributions of this paper include the following.

1. We review the characteristics of various existing emergency communication solutions, considering their technical and deployment complexity, and service recovery time. Additionally, we highlight the advantages of our work in relation to these characteristics.
2. We detail the design and deployment of a broadband mobile communications network architecture that allows access for users affiliated with any telecom operator. The architecture ensures independent support for communication services while minimizing interference with existing telecom operator networks.
3. We evaluate the effectiveness of the proposed home-independent broadband mobile communication network architecture. The numerical results demonstrate that our method provides good coverage, round-trip time, reference signal received power, and signal-to-interference plus noise ratio.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work and distinguishes our work from existing studies. The proposed UAV-assisted home-independent broadband mobile communication network is described in Sect. 3. Subsequently, Sect. 4 presents the corresponding performance evaluation, demonstrating the feasibility of the proposed network architecture and working principle. Finally, the concluding remarks and future work are discussed in Sect. 5.

2 Related Work

This section provides an overview of current emergency communication network architectures, including multi-base station, inter-network roaming, and multi-operator core network solutions. Moreover, it includes a table highlighting the distinctions between our research and existing methodologies. The detailed description is as follows.

2.1 Multiple Base Station Solution

The most direct solution is to deploy multiple sets of emergency base stations in disaster areas, each belonging to a different telecommunication operator [10]. The advantage of this solution is its simplicity in network design; however, practical experience has revealed the following problem.

1. Emergency management departments must coordinate base station equipment, satellite communication links, and core network access permissions with various telecommunications operators. The involvement of different departments within these operators complicates the coordination efforts, thereby hindering the ability to restore communication in disaster-affected areas efficiently.
2. The payload weight limitations of tethered UAVs limit them to carrying only a single set of base station equipment. The complex and narrow terrain of many disaster sites makes the deployment of multiple UAVs challenging due to safety concerns, thus posing significant challenges to implementing this solution.
3. The integration of airborne base stations requires all telecommunications operators to undertake intricate configuration tasks within their core networks.
4. Deploying multiple sets of base stations results in substantial costs.

2.2 Inter-network Roaming Solution

Inter-network roaming service enables mobile phone users to continue using mobile communication services even when moving to areas not covered by their home operator's network [11, 12].

Using the inter-network roaming solution, deploying one base station at the disaster relief site can provide emergency communication services to subscribers registered with multiple telecom operators.

However, implementing inter-network roaming among multiple domestic operators presents significant challenges. The primary reasons are as follows.

1. In most regions, the signal coverage of base stations from various domestic operators overlaps, leading to mobile phones often roaming between different networks as they move. This increases signaling traffic and disrupts mobile services by altering the signal strength received from these base stations.

Table 1. The comparison of our work with existing research

Solution	Technical complexity	Deployment complexity	Service recovery speed	Cost
Multiple base stations solution	Simple to design	High deployment complexity across multiple operators	Slow	High costs from numerous UAVs and base stations
Inter-network roaming solution	Moderate design difficulty & complex billing	Significant deployment challenges for mobile networks	Fast	Low construction, high operational costs
Multi-operator core network solution	High design complexity & complex wireless resources allocating	High deployment complexity	Slow	Low build, high operational costs
Our work	Moderate design difficulty, no billing, simple wireless resources allocating	Low deployment complexity	Fast	Low construction and operational costs

2. This constant roaming complicates the quality statistics for wireless access network services [13], posing challenges to operators' efforts in network optimization and improvement by removing a precise basis for these tasks.
3. The disparity in package plans and billing methods among different operators adds complexity to the existing billing systems and inter-network settlements stemming from inter-network roaming.
4. The restrictions in billing modes restrict roaming subscribers to accessing only a basic range of services, primarily limited to basic communication functions.

2.3 Multi-operator Core Network Solution

Using the multi-operator core network (MOCN) solution [14, 15], one set of base station is connected to the core networks of multiple telecom operators. Therefore, deploying a single base station in the rescue area can provide emergency communication services to subscribers registered with multiple operators.

Although this solution efficiently utilizes base station and carrier frequency resources, it still has room for improvement.

1. There is a significant level of interdependence among operators, resulting in elevated coordination and management expenses across multiple entities.
2. Allocating air interface wireless resources through a resource allocation strategy presents a complex challenge, making equitable distribution of these resources among various operators difficult.

3. Given that a single base station is concurrently connected to the core networks of multiple operators, each operator must adjust the configuration data of their respective core networks.
4. Emergency management departments must negotiate satellite communication links and core network access permissions with several telecommunications operators. The involvement of different departments within these operators complicates the coordination efforts, hindering the rapid restoration of communication in disaster-stricken areas.

To mitigate the limitations identified in existing research, we propose a novel broadband communication network architecture in this paper. This architecture overcomes the previously mentioned limitations by enabling subscribers affiliated with various telecommunications operators to access a single set of base stations, facilitating communication services for all subscribers in disaster-affected areas. Furthermore, this network architecture eliminates the need to coordinate satellite communication links and core network access permissions with multiple operators and the necessity for complex network configurations. The comparison of the proposed scheme with existing work is provided in Table 1.

However, our proposed architecture also needs expensive satellite bandwidth. In further studies this problem should be solved partly.

3 The UAV-Assisted Home-Independent Broadband Mobile communication Network Framework

3.1 Network Architecture

The primary application of the (tethered UAV-assisted) home-independent broadband mobile communication network framework is the broadband emergency communication network, which revolves around creating a dedicated emergency mobile core network. With a pair of optical fibers in the tether cable the airborne base stations within the rescue operation area connect to this dedicated core network, thus establishing a broadband emergency communication network. With an electricity cable in the tether cable the ground equipment supplies power to the UAV and base station carried by it. The architecture of the proposed communication network is illustrated in Fig. 1. This figure showcases a dedicated emergency communication core network, distinguished by a set of primary technical features that streamline the core network's structure.

1. The network enables interconnection with the mobile communication networks of various domestic operators.
2. It establishes selective inter-network roaming agreements with these operators, focusing primarily on user authentication protocols and signaling. This arrangement allows subscribers registered with other operators to roam into this network.
3. The network eliminates the need for billing functionality.

The wireless frequencies are acquired temporarily from telecommunications operators for the airborne base stations deployed over disaster-affected areas.

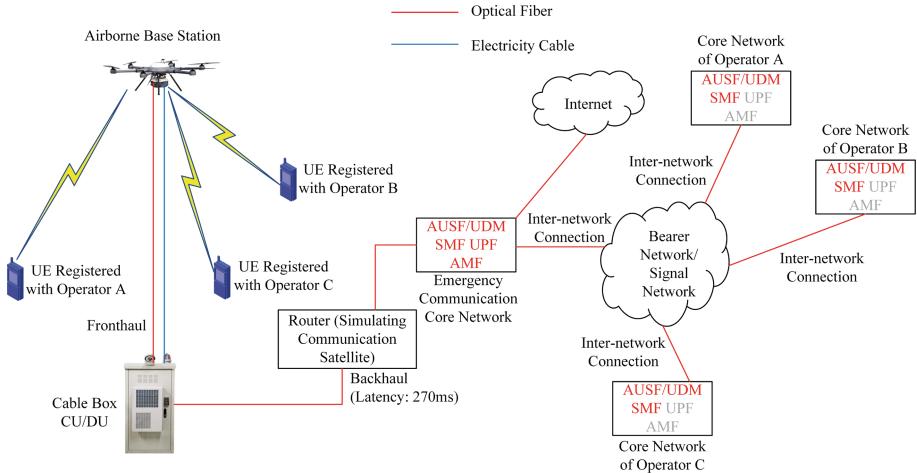


Fig. 1. The architecture of the UAV-assisted home-independent broadband emergency communication network

3.2 Working Principle and Service Process

The essence of the (UAV-assisted) home-independent broadband mobile communication network is its capacity to attain partial roaming signaling interoperability with the networks of other operators [16, 17]. The proposed network architecture is illustrated in Fig. 2.

In addition, the signaling process for UEs to access the proposed emergency communication network is described in Fig. 3.

As shown in Fig. 2 and 3, an UE registered with any one of other telecom operators sends access request to the home-independent broadband mobile communication network rather than to its home network which was damaged in disaster area, then the request is passed to UE's home operator network. The response signal from UE's home network's AUSF(hAUSF) is also passed to the UE via the home-independent broadband mobile communication network. Subscribers from various operators can access the network through Local Breakout (LBO) roaming, employing the network's Visited User Plane Function (vUPF) without depending on their home network infrastructure.

To enhance the clarity of our work, we have compiled a list of key abbreviations in Table 2.

Figure 3 also outlines the roaming process for UEs transitioning from another telecom operator to the home-independent broadband mobile communication network, enabling subscribers to access various communication services offered by the network.

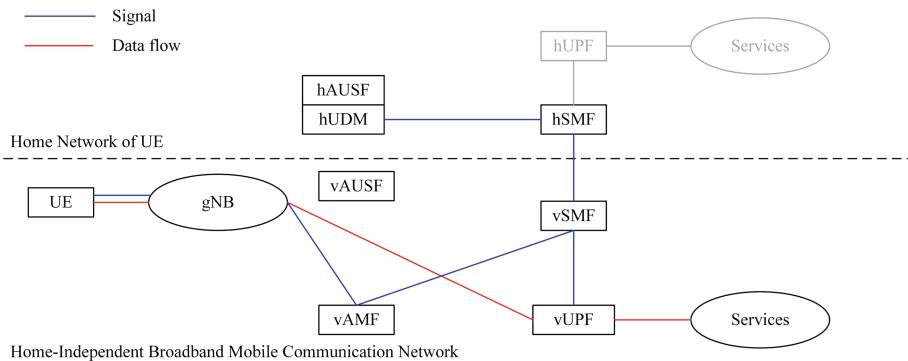


Fig. 2. The network architecture of UAV-assisted home-independent broadband emergency communication

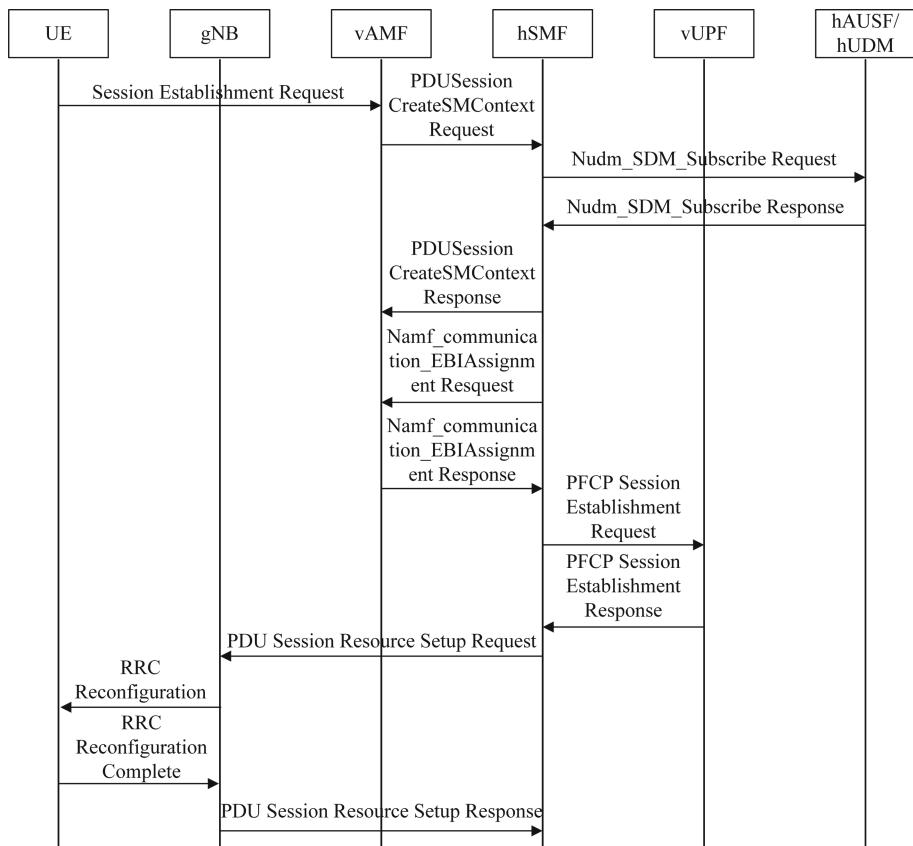


Fig. 3. The signaling process for UEs to access UAV-assisted home-independent broadband mobile network

Table 2. The important abbreviations used in our work

Abbreviation	Description & Definition
UE	User equipment, usually mobile devices
gNB	5G Base Station
AMF	Access and Mobility Management Function
SMF	Session Management Function
AUSF	Authentication Server Function
UDM	Unified Data Management
UPF	User Plane Function

4 Performance Evaluation

4.1 Simulation Settings

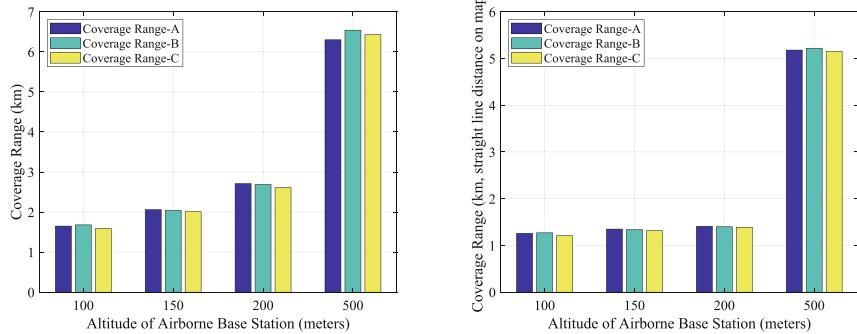
To evaluate the efficacy of our proposed solution, we executed a comprehensive suite of simulation tests encompassing coverage, PING success rate, round-trip delay, and wireless signal strength. The airborne base station's operational frequency band spans 1755–1785 MHz for uplink and 1850–1880 MHz for downlink, tested across mobile phones registered with operators A, B and C, respectively. The UAV, equipped with the base station, maintained hovering heights of 100, 150, 200, and 500 m.

4.2 Simulation Result

Firstly, we measured the simulation results of network coverage with UEs registered with three different operators, as shown in Fig. 4. Specifically, the data on the network coverage range, derived from the road test, is depicted in Fig. 4a, while Fig. 4b illustrates the straight-line distance from the base station's projected ground point to the coverage area's edge. We depict data from UEs registered with three distinct operators using three unique colors. Simulation statistics reveal that optimal performance is achieved by configuring the UAV altitude to 500 m.

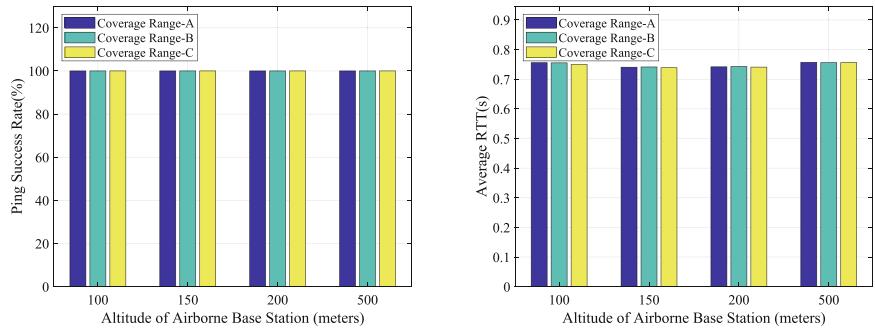
Subsequently, we measured the packet delivery rate and round-trip time by sending ping operations from the UE to the emergency rescue service server. Figure 5a illustrates the success rate of these pings via UEs, while Fig. 5b depicts the round-trip time. We observed that UEs registered with three distinct operators achieved a 100% PING success rate at various UAV altitudes. Furthermore, the corresponding PING round-trip times were consistently low, approximately 0.7 s.

Finally, we measured the wireless signal strengths of UEs registered by three different operators at different UAV heights, including reference signal receiving power (RSRP) and signal-to-interference-plus-noise ratio (SINR). Our statistics



(a) The comparison of coverage (road test). (b) The comparison of coverage radius.

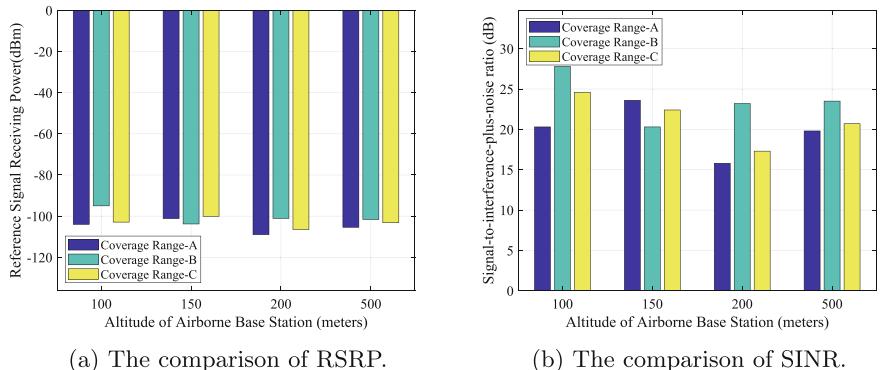
Fig. 4. The comparison of coverage measured with UEs registered with operator A, B, C



(a) The comparison of PING success rate.

(b) The comparison of PING RTT.

Fig. 5. The comparison of PING operation results measured with UEs registered with operator A, B, C



(a) The comparison of RSRP.

(b) The comparison of SINR.

Fig. 6. The comparison of wireless signal strengths measured with UEs registered with operator A, B, C

show that the RSRP performance of the three different operators is roughly the same at different UAV altitudes, as shown in Fig. 6.

The above simulation results demonstrate the effectiveness of our proposed network architecture, showing good performance in signal coverage, packet transmission rate, round-trip time, and signal strength, for all UEs registered with every telecom operator.

5 Conclusion

This paper introduces a novel UAV-assisted network architecture aimed at addressing the challenges in disaster-affected regions, such as limited base station infrastructure, coordination complexities among telecom operators and emergency services, and the fair distribution of resources among providers. Our findings highlight the method's effectiveness in ensuring signal coverage, packet delivery rate, round-trip time, and signal strength for UEs registered with every telecom operator. The communication performances shown on UEs registered with all telecom operators are good and almost consistent.

Future studies will focus on the deployment of 5G technology in disaster recovery, optimizing satellite bandwidth through 5G's UPF at disaster sites will be a key area of investigation [18]. This will minimize dependency on satellite links for non-critical traffic, potentially increasing communication capacity at disaster sites.

Acknowledgements. This work is supported in part by Major Research Project of National Natural Science Foundation of China under Grant 92267110.

References

1. Noorwali, A., Javed, M.A., Khan, M.Z.: Efficient UAV communications: recent trends and challenges. *Comput. Mater. Continua* **67**(1), 1–14 (2021)
2. Yin, M., Li, W., Feng, L., Peng, Yu., Qiu, X.: Emergency communications based on throughput-aware D2D multicasting in 5G public safety networks. *Sensors* **20**(7), 1901 (2020)
3. Saad, W., Bennis, M., Chen, M.: A vision of 6G wireless systems: applications, trends, technologies, and open research problems. *IEEE Netw.* **34**(3), 134–142 (2019)
4. Debnath, S., Arif, W., Roy, S., Baishya, S., Sen, D.: A comprehensive survey of emergency communication network and management. *Wireless Pers. Commun.* **124**(2), 1375–1421 (2022)
5. AFM Shahen Shah: Architecture of emergency communication systems in disasters through UAVs in 5G and beyond. *Drones* **7**(1), 25 (2022)
6. Wu, Q., Xu, J., Zeng, Y., Ng, D.W.K., Al-Dhahir, N., Schober, R., Swindlehurst, A.L.: A comprehensive overview on 5G-and-beyond networks with UAVs: from communications to sensing and intelligence. *IEEE J. Sel. Areas Commun.* **39**(10), 2912–2945 (2021)

7. Dicandia, F.A., Fonseca, N.J., Bacco, M., Mugnaini, S., Genovesi, S.: Space-air-ground integrated 6G wireless communication networks: a review of antenna technologies and application scenarios. *Sensors* **22**(9), 3136 (2022)
8. Liu, F., Cui, Y., Masouros, C., Xu, J., Han, T.X., Eldar, Y.C., Buzzi, S.: Integrated sensing and communications: toward dual-functional wireless networks for 6G and beyond. *IEEE J. Selected Areas Commun.* **40**(6), 1728–1767 (2022)
9. Wei, T., Feng, W., Chen, Y., Wang, C.-X., Ge, N., Jianhua, L.: Hybrid satellite-terrestrial communication networks for the maritime internet of things: key technologies, opportunities, and challenges. *IEEE Internet Things J.* **8**(11), 8910–8934 (2021)
10. Kimura, T., Ogura, M.: Distributed Collaborative 3D-deployment of UAV base stations for on-demand coverage. In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1748–1757. IEEE (2020)
11. Zhao, X., Hu, C., Mao, C., Li, P., Yu, J., Xie, W.: Field trials for 5G system roaming as multi-operator network sharing scheme—user mobility problems and solutions. In: *2022 IEEE 5th International Conference on Electronics Technology (ICET)*, pp. 949–953. IEEE (2022)
12. Mafakheri, B., Heider-Aivet, A., Riggio, R., Goratti, L.: Smart contracts in the 5G roaming architecture: the fusion of blockchain with 5G networks. *IEEE Commun. Mag.* **59**(3), 77–83 (2021)
13. He, Z., Li, J., Wu, F., Shi, H., Hwang, K.S.: DeRL: coupling decomposition in action space for reinforcement learning task. *IEEE Trans. Emerg. Topics Comput. Intell.* (2023)
14. Zhao, X., Hu, C., Li, Z.: Multi-operator radio access network sharing for 5G SA network design and laboratory test. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 187–193. IEEE (2021)
15. Zhang, Z., Li, Z., Chen, J., et al.: Research on 5G network co-construction and sharing technology. *Appl. Electron. Techn.* **46**(4), 1–5 (2020)
16. Rischke, J., Sossalla, P., Itting, S., Fitzek, F.H., Reisslein, M.: 5G campus networks: a first measurement study. *IEEE Access* **9**, 121786–121803 (2021)
17. Corici, M., Chakraborty, P., Magedanz, T., Gomes, A.S., Cordeiro, L.S., Mahmood, K.: 5G non-public-networks (NPN) roaming architecture. In: *2021 12th International Conference on Network of the Future (NoF)*, pp. 1–5. IEEE (2021)
18. Prados-Garzon, J., Ameigeiras, P., Ordóñez-Lucena, J., Muñoz, P., Adamuz-Hinojosa, O., Camps-Mur, D.: 5G non-public networks: standardization, architectures and challenges. *IEEE Access* **9**, 153893–153908 (2021)



FPTSF: A Failure Prediction of Hard Disks Based on Time Series Features Towards Low Quality Dataset

Xiaoyu Lu¹ Chenfeng Tu², Hongzhang Yang¹ Jiangpu Guo³, and Hailong Sun⁴

¹ Tianjin University of Technology, Tianjin 300384, China
yanghongzhang@email.tjut.edu.cn

² Haojing Cloud Computing Technology Corporation, Nanjing 211153, China
³ Roycom Information Technology Corporation, Tianjin 301721, China
⁴ China Electronics System Technology Corporation, Beijing 100036, China

Abstract. Hard disk failures cause data loss, reducing storage system reliability. While many machine learning models predict hard disk failures, they often neglect temporal characteristics and rely on high-quality datasets like Backblaze. In real-world industrial applications, data may be missing or of low quality. Current models perform poorly with such datasets. To address this, we propose a prediction technique tailored for low-quality datasets. We create a low-quality dataset, Backblaze-, by deleting 10% to 80% of data from the original Backblaze dataset. We introduce time series features like the Absolute Sum of First Difference (ASFD) to highlight fluctuations in hard disk SMART data, enhancing the distinction between positive and negative samples. Our approach achieves near-original dataset performance, with a TPR of 86.7%, an AUC score of 0.93, and an f1-score of 0.89, predicting failures 9.75 days in advance and maintaining an FPR below 0.01% even with 80% data missing.

Keywords: Failure prediction · Low quality data · Time series feature · Light Gradient Boosting Machine (LightGBM) · Proactive Reliability Assurance

1 Introduction

Hard disks are essential storage devices in computers and critical for data centers. Failures, which account for 78% of hardware replacements, are frequent in large-scale systems like high-performance computing and internet services [1]. These failures can lead to irreversible consequences, compromising data center reliability. Early prediction of failures can mitigate data loss risks and reduce recovery costs significantly.

Self-Monitoring Analysis and Reporting Technology (SMART) [2], introduced in the 1990s, monitors operational metrics like read/write counts and error rates in hard disks. Traditional threshold-based methods compare current SMART attributes with preset thresholds. Alerts are triggered if attributes exceed thresholds, achieving a TPR of 3% to 10% with an FPR of 0.1%. Researchers have explored several techniques over decades to

enhance failure prediction accuracy using machine learning and deep learning methods, including LSTM [3], GAN [4], RGF [5], and TCNs [6]. However, these models typically rely on high-quality datasets like Backblaze, limiting their applicability in environments with low-quality data.

In industrial applications, hard disk SMART data collection, transmission, and storage can result in partial data loss due to errors like sensor and transmission errors, device failures, and intentional data corruption by maintenance engineers. Ensuring reliable prediction performance with low-quality datasets is crucial for the practical application of hard disk failure prediction technologies.

This paper proposes FPTSF: A Failure Prediction of hard disks based on Time Series Features towards low quality dataset to address this. The main contributions of this paper are as follows:

1. This paper constructs and open-sources a low-quality SMART dataset named Backblaze-. Using SMART data from ST4000DM000 model hard drives released by Backblaze in 2022 and 2023, we deleted 10%~80% of the 2022 data. The data was preprocessed by resetting labels, selecting features, filling missing values, and adding time series features.
2. This paper addresses the temporal nature of hard disk data by employing a 10-day time window. Failing hard disks were labeled as 1 based on data from the last 10 days before imminent failure. Healthy hard disks are labeled as 0 using data from the first 10 days, which generally have the lowest likelihood of failure.
3. This paper finds that certain SMART attributes exhibit significant differences between healthy and failing disks. Therefore, we constructed time series statistical features for each SMART attribute to reflect absolute fluctuations between adjacent observations. Experimental results indicated that these time series features improve model accuracy and enable earlier prediction of hard disk failures.

The chapters of this paper are as follows: Introduction, Construction of Low-Quality Dataset, Time Series Features, Hard Disk Failure Prediction Model FPTSF, Experiments, Related Work, and Conclusion.

2 Construction of Low-Quality Dataset

This paper uses the high-quality Backblaze dataset [7] in 2022 and 2023, sampled daily with no data loss. The initial dataset consists of SMART data from ST4000DM000 model hard drives.

Table 1. Initial Dataset

Year	Total Hard Disks	Healthy Disks	Failed Disks
2022	18,611	17,978	633
2023	18,238	17,678	560
Total	36,849	35,656	1,193

As shown in Table 1, the initial dataset includes 36,849 hard disks, with 35,656 healthy and 1,193 failed disks. Data preprocessing in this paper involves data truncation and label resetting, feature selection, random data deletion, and missing value imputation:

1. Data truncation and label resetting. As the initial dataset only assigns a label of 1 to failed disks on the day of failure, one of the key objectives of this paper is to predict whether a hard disk will fail within future time windows, necessitating the reassignment of labels for failed disks. The paper truncates hard disk data within a 10-day time window to ensure reliability. For failed disks, it selects data from the last 10 days and labels it as 1, indicating imminent failure. This relabeling addresses the original dataset's limitation of labeling only on the failure day. For healthy disks, unlike the method used by Han S [8], data from the first 10 days is selected and labeled as 0, as this period is the most stable and least prone to failure.
2. Feature selection: Remove dataset features that are entirely null or have 0 variances, as these do not contribute to classification. SMART attributes IDs after feature selection are as follows: *Raw*: #12, #188, #192, #199, #240, #241, #242; *Normalized*: #10; *Raw & Normalized*: #1, #4, #5, #7, #9, #183, #184, #187, #189, #190, #191, #193, #194, #195, #197, #198.
3. Random deletion of data. We used Numpy's `np.random.choice()` to randomly sample and delete 10%~80% of the original data, creating the low-quality dataset Backblaze-. This dataset is open-sourced (<https://github.com/cccatt-best/Backblaze-.git>) and includes nine parts: *all_data*, *drop_data_10*, *drop_data_20*, *drop_data_30*, *drop_data_40*, *drop_data_50*, *drop_data_60*, *drop_data_70*, and *drop_data_80*.
4. Missing value imputation. In this study, missing values were imputed using the mode of the attribute column where the missing values occurred.

3 Time Series Features

Current hard disk failure prediction methods often rely on same-day SMART data, missing the gradual process of failure where certain attributes change abruptly. To capture abrupt changes, we used the Moving Average (MA) in the previous work, and the effect was not ideal. In this paper, the sliding window method is used to add the Absolute Sum of First Difference (ASFD) of time series features to the data set, assisted by adding Complexity Invariant Distance (CID) for observation and comparison. They are used to reflect the absolute fluctuation between adjacent observations of SMART data and the complexity of time series.

3.1 Absolute Sum of First Difference

The Absolute Sum of First Difference (ASFD) quantifies variations between successive values in a time series. It sums the absolute differences between each element and its predecessor, calculated as shown in Eq. (1). ASFD values reflect the magnitude of changes in the series, with higher values indicating greater variations.

$$y = i = 1, \dots, n - 1 |x_i + 1 - x_i| \quad (1)$$

This paper demonstrates the effectiveness of ASFD through four time series:

1. A time series of length 100 with all values set to 0.
2. Based on time series 1, one point is randomly selected and set to 10.
3. Based on time series 1, one point is randomly selected and set to 100.
4. Based on time series 1, two points are randomly selected and set to 10.

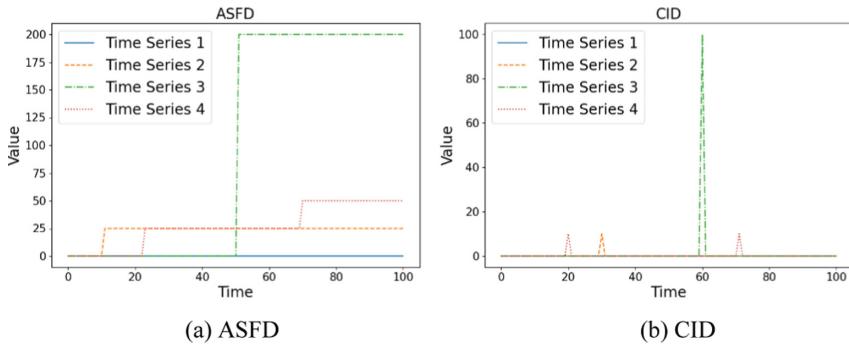


Fig. 1. Characteristic Values of Different Time Series

As shown in Fig. 1a, the ASFD feature can effectively assess the absolute fluctuations between adjacent observations in a time series and detect abrupt changes within the series.

1. Adding abrupt changes to a time series increases the ASFD.
2. The further the values deviate from normal, the larger the ASFD of the time series.
3. The more frequent the abrupt changes, the larger the ASFD of that time series.
4. The ASFD magnitude accumulates with the number of abrupt changes.

3.2 Complexity Invariant Distance

Complexity Invariant Distance (CID) is a metric for comparing the similarity between time series data, designed to overcome traditional distance metrics' limitations with complex data distributions. Its effectiveness is demonstrated using the four simulated time series discussed in Sect. 3.1.

The experimental results, as shown in Fig. 1b, demonstrate that the CID feature effectively evaluates the complexity of time series as well as the occurrence of abrupt changes within them.

1. Adding abrupt changes to a time series increases the CID.
2. The further the values deviate from normal, the larger the CID of the time series.
3. The more frequent the abrupt changes, the greater the CID of the time series.

4 Hard Disk Failure Prediction Model

4.1 Hard Disk Failure Prediction Process

The hard disk failure prediction process in this paper mainly includes the following steps:

1. Adding Time Series Features. After preprocessing the 2022 dataset, ASFD and CID features were added to each SMART attribute.
2. Splitting Training and Testing Sets. The data was stratified to ensure a proportional representation of positive and negative samples, then randomly divided into 80% training and 20% testing sets.
3. Model Training. The LGB model by Microsoft [9] was used for training the hard disk failure prediction model with the preprocessed training set. Other algorithms were also tested for comparison.
4. Model Tuning. OPTUNA was used to find the optimal hyperparameters for the LightGBM algorithm, allowing for model tuning.
5. Hard Disk Failure Prediction. The optimal model was used to predict disk failures by inputting test set data and obtaining disk-level predictions.
6. Model Evaluation. Using the disk-level predictions from the test set and the corresponding labels, the model's performance was evaluated with AUC, f1-score, and TPR (where $FPR < 0.1\%$).
7. Calculation of Days Predicted in Advance. After obtaining the trained high-performance model, the unreset labeled data of all failed hard disks from 2023 was used as the test set to calculate the days the model predicted the failures in advance.

4.2 Model Hyperparameter Tuning

For hyperparameter tuning, many researchers use the brute-force GridSearch method [10]. While straightforward, it is notably time-consuming. In contrast, Bayesian framework-based tuning methods such as HyperOPT [11] and OPTUNA [12] are more efficient and powerful. OPTUNA, being lightweight and extremely fast, was chosen as the tuning method in this study.

Typically, the hyperparameters of tree-based models can be divided into four categories: 1) Parameters affecting the structure and learning of the decision tree. 2) Parameters influencing training speed. 3) Parameters improving accuracy. 4) Parameters preventing overfitting.

These categories often overlap, and improving one may compromise another. Manual tuning is labor-intensive and time-consuming, making it difficult to achieve optimal balance. Optuna automates the search for balanced parameter combinations, enhancing overall performance.

5 Experiments

5.1 Model Evaluation Metrics

Common metrics like accuracy and precision are inadequate for hard disk failure prediction due to dataset imbalance, where negative samples outnumber positives. This study focuses on TPR, FPR, AUC score, f1 score, and Days to Predict Failure (DPF) metrics. DPF was calculated by extracting all-time data from 2023 for failed disks without resetting labels and testing them with the trained model.

5.2 Experiment on the Impact of Dataset Quality

To verify the effect of dataset quality on fault prediction, this experiment was conducted based on the initial 2022 dataset after data truncation, label resetting, and feature selection. Subsequently, data ranging from 10% to 80% were randomly deleted to construct low-quality datasets labeled as Backblaze-. These datasets were then used to train LGB models for comparison. The time window size was set to 10 days.

Table 2. Experimental Results of Different Quality Datasets

Dataset	TPR	FPR	AUC score	f1-score
all_data	0.9984	0.0001	0.9992	0.9984
drop_data_10	0.9927	0.0001	0.9963	0.9947
drop_data_20	0.9661	0.0002	0.9830	0.9804
drop_data_30	0.9444	0.0004	0.9720	0.9650
drop_data_40	0.9169	0.0011	0.9579	0.9404
drop_data_50	0.8992	0.0020	0.9486	0.9188
drop_data_60	0.8427	0.0031	0.9198	0.8727
drop_data_70	0.7540	0.0054	0.8743	0.7897
drop_data_80	0.6548	0.0236	0.8156	0.5598

The experimental results, as shown in Table 2, indicate a noticeable decrease in the performance of the LGB model across all four evaluation metrics as more data is deleted. TPR decreased from 0.9984 to 0.6548, the AUC score decreased from 0.9992 to 0.8156, and the f1-score decreased from 0.9984 to 0.5598. Therefore, we can conclude that higher dataset quality leads to a higher upper limit of accuracy in predicting hard disk failures, whereas lower dataset quality leads to a lower limit.

5.3 Stability Experiment of Low-Quality Dataset

To verify the low-quality dataset's reliability, we conducted experiments using the 2022 initial dataset after truncating data, resetting labels, and selecting features. Ten different random seeds were used to create ten low-quality datasets by randomly deleting 50% of the data using various methods. An LGB model was trained with a 10-day time window for comparison.

Table 3 displays experimental results across 10 datasets: average TPR of 0.8488, AUC of 0.9240, and f1-score of 0.9059, each with standard deviations under 0.01. These findings confirm the effectiveness and stability of constructing our low-quality dataset, Backblaze-.

5.4 Model Comparison and Selection

To evaluate the effectiveness of hard disk failure prediction across different models, we compared the training results on various datasets: drop_SMART (80% data

Table 3. Experimental Results of Randomly Deleting 50% of Data

Dataset	TPR	FPR	AUC score	f1-score
drop_50_1	0.8629	0.0010	0.9310	0.9126
drop_50_2	0.8508	0.0012	0.9248	0.9029
drop_50_3	0.8323	0.0009	0.9157	0.8951
drop_50_4	0.8524	0.0007	0.9258	0.9100
drop_50_5	0.8444	0.0009	0.9217	0.9030
drop_50_6	0.8556	0.0007	0.9275	0.9119
drop_50_7	0.8508	0.0009	0.9249	0.9064
drop_50_8	0.8500	0.0007	0.9247	0.9094
drop_50_9	0.8363	0.0009	0.9177	0.8978
drop_50_10	0.8524	0.0008	0.9258	0.9096
Average	0.8488	0.0009	0.9240	0.9059
STDEV	0.0090	0.0002	0.0045	0.0060

removed), filled_SMART (missing values filled), SMART+CID, SMART+ASFD, and SMART+ASFD+CID. Models used included LGB, XGBoost, CatBoost, NGBoost, AdaBoost and LSTM. Since NGBoost, AdaBoost, and LSTM require complete datasets, results for the drop_SMART dataset are not provided for these models.

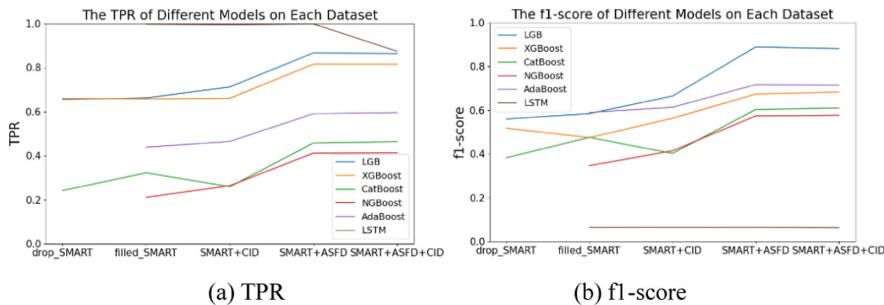
**Fig. 2.** The Effect of Different Models on Each Dataset

Figure 2 illustrates the performance of different models across various datasets. The horizontal axis represents five different datasets, while the vertical axis indicates the scores of different metrics for each model under the respective dataset. In (a), we observe the effect of different models on TPR. In (b), the influence of different models on the f1-score is shown. From the graph, we can discern that:

1. The effectiveness of the five models varies across different metrics in hard drive failure prediction, with the overall results showing that the LGB model performs the best in terms of TPR and f1-score.

2. Both the LGB and XGBoost models demonstrate good performance in TPR metrics for hard drive failure prediction. In comparison, the CatBoost and NGBoost models exhibit less favorable results.
3. While AdaBoost performs poorly in TPR metrics, it achieves relatively high scores in the f1-score metric.
4. Although the LSTM model can achieve high TPR, it is not suitable for low-quality data sets, and the f1-score is very low.
5. After deleting 80% of the data, the XGBoost model achieves the highest TPR score, reaching 0.6605. However, when the dataset with deleted entries is filled with mode values, the LGB model surpasses XGBoost in TPR. With the addition of time series indicators, the LGB model still performs the best, with the highest TPR reaching 0.8669.
6. Adding CID to the filled dataset negatively impacts the CatBoost model but positively correlates with other models. Including ASFD significantly improves TPR for all models, proving its effectiveness in hard drive failure prediction. Using both ASFD and CID yields a TPR similar to ASFD alone, indicating ASFD's greater suitability for these scenarios.

5.5 Experiment on Model Effectiveness and Days to Predict Failure

To verify our hard disk failure prediction model, we used a real dataset collected by the Nankai University-Baidu Joint Laboratory [13], processed it with the same low-quality quantization method, and conducted experiments. The results are as follows.

Table 4. Experimental Results of Real Datasets

Dataset	TPR	FPR	AUC score	f1-score
Nankai	0.9993	0.0319	0.9837	0.9990

From the above results, we can see that our hard disk failure prediction model for low-quality datasets also achieves ideal prediction results on real datasets. This proves the model's effectiveness and generalizability.

As mentioned earlier, before training the model, we reset the labels of the ST4000DM000 hard disk dataset publicly released by Backblaze in 2022. We set the time window to 10 days, ensuring that failing hard disks can be detected before failure occurs. In this experiment, we used the LGB model and extracted all the time data of the failing ST4000DM000 hard disks publicly released by Backblaze in 2023 as the test set. The ideal result would be that all data are predicted as 1. The actual results are shown in Table 4. The test set includes 560 failing disks, with a total data volume of 87,995.

Table 5 shows the model predictions on the test set. The DPF for different levels of data deletion is consistently above 9.5 days, with a variance of less than 0.06, demonstrating stable results. Our model successfully predicts failing disks with up to 9.75 days lead time, even when 80% of the data is deleted, highlighting its crucial role in ensuring data reliability in industrial applications.

Table 5. Test Set Prediction Results

Removal Rate	Positive Predictions	Probability	DPF(days)
0	86,606	0.9842	9.84
50%	86,736	0.9857	9.86
80%	85,824	0.9753	9.75
STDEV		0.0056	0.0561

6 Related Work

Over the past few decades, numerous scholars have proposed methods based on machine learning and deep learning to improve the accuracy of hard drive failure prediction:

Tianming J. focused on minimizing the economic cost of hard drive failure recovery [14]. They introduced the MCTR (Mean-Cost-To-Recovery) metric and employed a threshold-moving approach to optimize results. Evaluation across three datasets showed an 86.9% reduction in costs compared to passive fault-tolerant techniques.

Instead of developing new models, Shujie H. focused on preprocessing hard drive datasets [15]. In real-world environments, data often has issues like incorrect labels, missing samples, and complex failure types. Shujie H. introduced the RODMAN pipeline to optimize datasets before model training. Evaluations combining SMART logs, system logs, and failure events showed that RODMAN improved the accuracy of machine learning models across various datasets.

To address imbalanced samples in hard drive datasets, Yong X. used the FastTree algorithm to transform classification into a ranking problem [16]. Using one month of SMART data and system-level metrics from Microsoft Azure, they achieved an AUC score of 0.9, with a TPR of 41% at an FPR below 0.1%.

In summary, these methods have several main shortcomings: 1) Ignoring the temporal aspect of hard drive data: Most failure prediction methods treat each day's data as isolated samples, overlooking temporal information and early signs of instability in the gradual failure process. 2) Lack of higher-order feature exploration: Most researchers use raw SMART datasets, which don't capture abrupt changes. Incorporating higher-order time series features could improve prediction accuracy. 3) Poor applicability in industrial scenarios: Existing methods use high-quality datasets like Backblaze's, but real-world data often has missing values and imbalanced samples. Random dataset splitting for training and testing does not reflect actual data center application patterns.

7 Conclusion

In addressing poor-quality datasets in industrial settings, this paper proposes FPTSF, a time-series feature-based method for hard drive failure prediction tailored to such conditions. Using the 2022 Backblaze public dataset, we systematically reduced data by 10% to 80% to create the low-quality dataset Backblaze-. Following data preprocessing, we introduced ASFD and CID as time-series features. Experimental results show that

both ASFD and CID notably improve hard drive failure prediction, with ASFD showing particular effectiveness. Moreover, our experiments on the real dataset collected by Nankai University demonstrate the efficacy and generalizability of our hard disk failure prediction model for low-quality datasets.

Acknowledgments. This research was supported by a grant from the Tianjin Manufacturing High Quality Development Special Foundation (No. 20232185) and the Roycom Foundation.

References

1. Xu, S., Xu, X.: ConvTrans-TPS: a convolutional transformer model for disk failure prediction in large-scale network storage systems. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 1318–1323. IEEE (2023)
2. Allen, B.: Monitoring hard disks with SMART. *Linux J.* **117**, 74–77 (2004)
3. Coursey, A., Nath, G., Prabhu, S., et al.: Remaining useful life estimation of hard disk drives using bidirectional LSTM networks In: 2021 IEEE International Conference on Big Data, pp. 4832–4841. IEEE (2021)
4. Liu, Y., Guan, Y., Jiang, T., et al.: SPAE: lifelong disk failure prediction via end-to-end GAN-based anomaly detection with ensemble update. *Futur. Gener. Comput. Syst.* **148**, 460–471 (2023)
5. Gargiulo, F., Duellmann, D., Arpaia, P., et al.: Predicting hard disk failure by means of automatized labeling and machine learning approach. *Appl. Sci.* **11**(18), 8293 (2021)
6. Burrello, A., Pagliari, D.J., Bartolini, A., Benini, L., Macii, E., Poncino, M.: Predicting hard disk failures in data centers using temporal convolutional neural networks. In: Balis, B., et al. (eds.) Euro-Par 2020: Parallel Processing Workshops. Euro-Par 2020. LNCS, vol. 12480. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-71593-9_22
7. Backblaze Hard Drive Data and Stats. <https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data>
8. Han, S., Wu, J., Xu, E., et al.: Robust data preprocessing for machine-learning-based disk failure prediction in cloud production environments, pp.1–12. arXiv preprint [arXiv:1912.09722](https://arxiv.org/abs/1912.09722) (2019)
9. Ke, G., Meng, Q., Finley, T., et al.: Lightgbm: a highly efficient gradient boosting decision tree. *Adv. Neural Inform. Process. Syst.* **30** (2017)
10. Belete, D.M., Huchariah, M.D.: Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *Int. J. Comput. Appl.* **44**(9), 875–886 (2022)
11. Zhang, J., Wang, Q., Shen, W.: Hyper-parameter optimization of multiple machine learning algorithms for molecular property prediction using hyperopt library. *Chin. J. Chem. Eng.* **52**, 115–125 (2022)
12. Lai, J.P., Lin, Y.L., Lin, H.C., et al.: Tree-based machine learning models with optuna in predicting impedance values for circuit analysis. *Micromachines* **14**(2), 265 (2023)
13. Li, J., Stones, R.J., Wang, G., et al.: Hard drive failure prediction using decision trees. *Reliab. Eng. Syst. Saf.* **164**, 55–65 (2017)
14. Jiang, T., Huang, P., Zhou, K.: Cost-efficiency disk failure prediction via threshold-moving. *Concurr. Comput. Pract. Exper.* **32**(14), e5669 (2020)
15. Han, S., Wu, J., Xu, E., et al.: Robust data preprocessing for machine-learning-based disk failure prediction in cloud production environments. arXiv preprint [arXiv:1912.09722](https://arxiv.org/abs/1912.09722) (2019)
16. Xu, Y., Sui, K., Yao, R., et al.: Improving service availability of cloud systems by predicting disk error. In: 2018 USENIX Annual Technical Conference, pp. 481–494 (2018)



PMEMgreSQL: Embracing PostgreSQL with Persistent Memory

Xinyuan Sun^(✉), Ji Shi, Yinjun Han, and Zhenghua Chen

Nanjing Zhongxing New Software Company Limited, Nanjing, China
yaphetsun98@gmail.com

Abstract. Persistent memory (PMem) is a transformative technology that holds the potential to significantly enhance the performance of database systems like PostgreSQL (PG). The emerging persistent memory is promising for boosting database performance because of its byte-addressable persistence with high bandwidth and low latency. However, the integration of PMem with databases is not without its challenges, particularly when considering concurrent processing and Non-Uniform Memory Access (NUMA) architectures. We encounter two unique issues which significantly affect system performance when integrating PMem into PG. Firstly, unrestricted concurrent access to PMem can lead to bandwidth degradation, which negatively impacts overall system performance. Secondly, when data is accessed across different NUMA nodes, performance can suffer due to increased latency. In this case, we propose PMEMgreSQL, embracing PG with persistent memory while solving the two issues above effectively. We first design a PMem-aware parallelism control mechanism which keeps processes accessing PMem in a queued and controlled manner. This approach involves careful scheduling of read and write operations to PMem, preventing the processes from menacing the memory bandwidth. Then we present a cross-NUMA logging interleave strategy which allows parallel logging in PMem devices in different NUMA nodes. This strategy ensures that the logging operation is distributed efficiently, reducing the performance penalty associated with cross-NUMA memory access.

Keywords: Persistent Memory · Process Scheduling · Database System

1 Introduction

The emerging persistent memory brings opportunities for optimizing databases that PMem can be utilized as an expansion of limited DRAM [2,5] or as an alternative for low-speed disk devices [6,12]. PMem supports byte-addressable persistence with high bandwidth and low latency. However, there are some special hardware characteristics of PMem that hinders performance improvement, such as 256B internal blocks and the dramatic performance degradation for cross-NUMA access. Several works [1,3,4,8,11] have analyzed these observations and

proposed corresponding optimized data structures or mechanisms to further saturate the bandwidth of persistent memory.

When integrating PMem into the PG database system, we identify and analyze two distinct challenges: 1) We observe a performance decline due to excessive process contention on PMem. As depicted in Fig. 1, the write bandwidth of PMem begins to diminish rather than increase when the concurrency level surpasses eight threads. 2) There is a notable performance hit due to the separation of DRAM and PMem across different NUMA nodes. Figure 1 illustrates that when DRAM and PMem are not collocated within the same NUMA node, the PMem write bandwidth can drop by as much as 40%. To align CPU and PMem within the same NUMA node, it is preferable to also situate the DRAM used for data interchange within that same NUMA node.

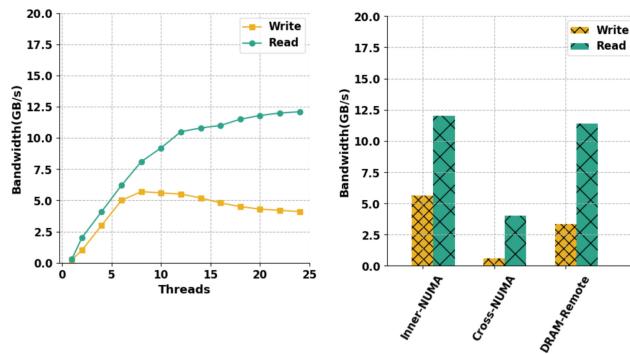


Fig. 1. PMem bandwidth under varying threads and configurations Inner-NUMA: Threads, PMem, and DRAM all in NUMA node 0 Cross-NUMA: PMem in NUMA node 0; Threads/DRAM in NUMA node 1 DRAM-Remote: PMem/Threads in NUMA node 0; DRAM in NUMA node 1

In this paper, we present our efforts to enhance the PG database system through the integration of PMem, leveraging its unique properties to optimize performance. Our contributions are as follows:

- * We develop a PMem-aware parallelism control mechanism, which employs a circular, lock-free data structure known as pList, to manage concurrent operations efficiently. Experiments show performance improvements of 18.87% on average compared with the original PG.
- * We introduce a priority scheduling algorithm designed to order processes based on a set of defined parameters, enhancing the system's latency and throughput. This algorithm has a considerable effect on helping and improving the control mechanism above.
- * We formulate a cross-NUMA logging strategy that interleaves logging operations to make optimal use of both local and remote PMem. Our strategy owns a noticeable advantage of 15.8% overall when comparing with the original PG in write operations.

2 PMEMgreSQL Design

In this paper, we present an upgraded version of the PG database system by migrating two key component types to PMem: in-memory components, including the B+ tree index and shared cache, as well as disk components, such as the log and heap storage. We introduce the following enhancements: 1) A PMem-aware parallelism control mechanism that is designed to manage concurrent operations effectively, taking full advantage of PMem's capabilities. 2) A process priority scheduling algorithm that intelligently sequences different processes to optimize resource utilization and system throughput. 3) A cross-NUMA logging interleave strategy that coordinates logging operations across NUMA nodes, ensuring efficient use of both local and remote PMem to maximize bandwidth utilization. These innovations are aimed at fulfilling the performance of PG through the strategic use of PMem, thereby providing a more efficient database system.

2.1 PMem-Aware Parallelism Control

We develop a fixed-size K -ring list, termed $pList$, which is instrumental in managing the maximum number of processes that can simultaneously write to a single PMem device. The dimension of $pList$ is determined empirically. For instance, in our experiment setup, which includes two PMem devices per NUMA node, we find that an optimal number of 8 write processes corresponds to the peak PMem bandwidth. The $pList$ is located in shared memory, ensuring visibility and accessibility to all PG processes. As shown in Fig. 2(a), each entry in $pList$ has three fields: a 1-bit *Flag*, a 17-bit *Pid* (process ID), and a 46-bit *Lease*. The *Flag* indicates whether the entry is in use. The *Pid* holds the lower 17 bits of the actual process ID, with a range that exceeds the default `pid_max`(32768) in Linux, and includes a reserve for potential `pid_max` adjustments. The total size of an entry is 64 bits. And a $pList$ of size 8 is 64 bytes, which corresponds to the size of a CPU cacheline. Memory allocation for these entries is also aligned to the cacheline to ensure optimal performance. The $pList$ is designed as a lock-free array, and the *Flag* of each entry is toggled using Compare-And-Swap(*CAS*) instructions, thereby completely avoiding the use of locking mechanisms.

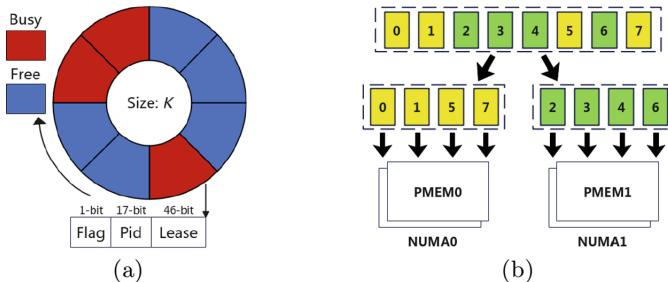


Fig. 2. Data Structure of $pList$ & Cross-NUMA Logging Strategy

Algorithm 1 outlines the fundamental steps for controlling parallelism in PMEMgreSQL. When a process wants to access the PMem, this process is placed into a waiting list, denoted as *wList*. As the transaction encounters the need to write, the process moves into the execution phase. A new *pList* item is then created, referred to as *new_item*, and the process operates initialization of this item. The *Flag* is initialized to 1, meaning the slot is in use (Phases 1–4). When the item's ready, the process then searches for an unoccupied slot within *pList* and invokes the *CAS* operation to claim the slot. Should *CAS* be successful, the process writes to PMem, and updates the *Lease* at regular intervals to indicate activity. If *CAS* fails, or the *pList* item is occupied, the process enters a retry loop, continuing to search for the next available slot. This incrementally increases its *wait_time* in *wList* to avoid infinite waiting (Phases 6–20). Furthermore, a monitor process checks the *pList* and if a *Lease* keeps unchanged for a long time, it considers the process crashed. The monitor process then resets the *Flag* to 0, freeing the slot. This algorithm ensures an orderly approach to concurrently access to PMem, optimizing the performance of PMEMgreSQL.

Algorithm 1: PMem-Aware Parallelism Control

```

1 wp = fork(); //Initialize a working process, wp.
2 wList.push_back(wp); //Initialize a waiting list, wList.
3 //Initialize a pList item, new_item.
4 new_item.initialize(flag=1, pid=get_pid(wp), lease=0)
5
6 for item in pList do
7   if item.flag == 0 then
8     ok = compare_and_swap(&item, item, new_item);
9     if ok == True then
10       //Placed successfully.
11       wp accesses PMem; pList[wp].lease++;
12     else
13       //Placed unsuccessfully.
14       wList[wp].wait_time++; continue;
15     end
16   else
17     //Look for the next free item.
18     wList[wp].wait_time++; continue;
19   end
20 end

```

2.2 Process Priority Scheduling

To establish a fair and efficient scheduling policy that prevents process starvation, we introduce an innovative process scheduling algorithm designed to

balance the average waiting time for each process. This algorithm takes PMem bandwidth as a dynamic threshold for its operation. Under conditions where PMem bandwidth utilization is relatively low, the scheduling mechanism remains inactive. This is because there is minor concurrent contention, it is not necessary to introduce a scheduling strategy.

Conversely, when the PMem bandwidth utilization escalates to high levels (e.g. exceeding 80%), the contention-aware process priority scheduling mechanism is activated. This mechanism steps in to manage the scheduling of waiting processes, ensuring a fair access of PMem and preventing any single process from being over-delayed. In this event, our scheduling algorithm calculate a priority score for each waiting process. The priority score is then used to fairly and efficiently allocate access to PMem, ensuring that the system's performance is optimized while maintaining equity among competing processes.

Equation 1, as illustrated, defines a priority scoring formula where T symbolizes the waiting time of a process, N denotes the amount of data that the process intends to write to PMem and U represents for user-defined weight. Normally, processes with higher priority get to use PMem first. However, if a process has been waiting for a long time, its score goes up in order to prevent it from being starved out. The algorithm also considers the impact of the data amount. If a process has an amount of data to write, it could take up PMem for a long time, making other processes wait longer. Settlements incur that the more data a process wants to write, the lower its priority score will be. This helps keep the system balanced and efficient by reducing the impact of long write operations on other processes.

$$P_{priority} = \frac{U \times T}{N} \quad (1)$$

Algorithm 2 delineates the foundational workflow of the process priority scheduling mechanism within PMEMgreSQL. PMEMgreSQL initiates the scheduling mode by setting the scheduling flag to True and assuming control of the $pList$. This signifies that the system is now prioritizing write access to PMem based on a calculated score rather than on a contention mode (Phases 1–2). Subsequently, the scheduling process evaluates and computes the priority score for each item within the waiting list, $wList$ (Phases 4–6). Once the scores are calculated, PMEMgreSQL proceeds to sort the $wList$ items in descending order based on their priority scores, ensuring that the highest-priority processes are positioned at the front of $wList$ (Phases 8–9). Finally, PMEMgreSQL grants writing privileges to the first K processes from the sorted $wList$, where K is the size of $pList$. By enabling these selected processes to write to PMem simultaneously, the system optimizes its performance while maintaining a fair scheduling policy (Phases 10–11). This priority scheduling algorithm is a pivotal component of PMEMgreSQL's approach to managing concurrent access to PMem, aiming to enhance overall system efficiency and ensure equitable resource allocation among competing processes.

Algorithm 2: Process Priority Scheduling

```

1 //Turn on the scheduling switch and take over the pList.
2 set_scheduling_flag(True);
3
4 for item in wList do
5   | item.priority = cal_priority(item); //priority = U * T / N
6 end
7
8 //Sort by the value of priority scores in descending order.
9 sort(wList.begin(),wList.end());
10 //Assign PMem access right to k processes.
11 assign_right(wList,k);

```

2.3 Cross-NUMA Logging Strategy

PMEMgreSQL employs a strategic approach in order to optimize resource allocation and maximize the utilization of PMem bandwidth. Memory allocation is first attempted within the local NUMA node, ensuring that the process, DRAM, and PMem are all within the same NUMA node. It is only when the local memory are depleted that the process allocate memory from a remote NUMA node. This careful management of resources is pivotal in fully harnessing the potential of PMem, thereby boosting the performance of the PG database system. In our work, we mainly focus on write operations for that the write performance is significantly low compared with read.

To further enhance system performance, we introduce a cross-NUMA logging interleave mechanism that enables parallel writing of logs to PMem devices across various NUMA nodes. Given that PMem devices are spread across different NUMA nodes and that cross-NUMA access tends to be less efficient, it is crucial for processes to access PMem devices within a single NUMA node. However, the bandwidth provided by a single PMem device is inherently limited. To address this, our mechanism capitalizes on the collective bandwidth of multiple cross-NUMA PMem devices by facilitating parallel log operations. The mechanism achieves this by strategically distributing log buffers across PMem devices in different NUMA nodes, allowing for concurrent log writing.

Before a log is written, the Write-Ahead Logging (WAL) location is pre-allocated within the log buffer, which is based on the current WAL location being written and the length of the WAL record to be inserted. We implement the log buffer with PMem. As illustrated in Fig. 2(b), the example shows that the log buffer is segmented into eight distinct parts (0–7). These parts are strategically placed in an interleaved pattern across two PMem devices. According to our algorithm, each part should assign PMem and complete writing on their processes' NUMA node (e.g. parts 0, 1, 5, and 7 are located on PMem device 0, while parts 2, 3, 4, and 6 are on PMem device 1). This distribution allows for parallel log-writing operations across different NUMA nodes, thereby improving throughput. Then *Flush* and *Fence* instructions are used to persist the log

in PMem. In the traditional PG system, transactions are only considered committed once their associated logs are persisted to disk. However, with PMEMgreSQL, this can be done by copying its redo-log to the PMem WAL buffer (after *Flush* and *Fence*), avoiding memory copy for one time. The logs are written in a partitioned manner, and the Log Sequence Number (LSN) is instrumental in maintaining the global sequence of these logs, providing a reliable order of transactions.

3 Evaluation

3.1 Experiment Setup

The operating system we use is CentOS Linux release 7.9.2009 (Core), equipped with 2 CPUs (Intel Xeon Gold 6230N CPU @ 2.30 GHz); 12 DRAMs (32G DDR4) and 4 PMems (Intel Optane DC 256GB Persistent Memory 100 Series). Sysbench supports tests of CPU, memory, file I/O, database benchmarking, etc. We operate OLTP experiments on Sysbench 1.0.20 using built-in test sets which includes 10 tables, with at least 1 million rows each. The test sets have some minute discrepancies after multiple rounds of Insert, Update, and Delete operations.

Given that the original PG operates with a default serial disk flushing mechanism, such as using a single BGWriter process, which does not fully harness the potential of parallel I/O, we allow the worker processes to complete I/O operations concurrently on their own, instead of relying on the BGWriter process. This is termed as the ‘PG-parallel’ version. The concurrency control algorithm introduced in this paper is implemented based on this upgraded version.

3.2 Parallelism Control Experiment

We conduct comparative experiments to validate the performance enhancement of the PG database without concurrency restrictions by establishing control groups with PG-origin and PG-parallel. As depicted in Fig. 3, our experiments focus on four key operations: Insert, Select, Update, and Delete.

During the Insert operation, which involves a high volume of random write, performance of PMEMgreSQL increases by 24.7% and 10.6% compared with PG-origin and PG-parallel. This is attributed to our PMem-aware parallelism control, which enhances the overall Write performance, resulting in a significant improvement in the Insert operation. While in the case of the Select operation, which is primarily read-intensive, the performance is essentially the same, because our work primarily focuses on optimizing the Write operations. The performance differences among the three versions are negligible (2.4% and 1.9%). The Update operation, reflecting a mixture of reads and writes on the disk, shows performance improvements lower than that of Insert, which corroborate with our results (15.7% and 7.4%). The Delete operation is similar to Insert but with smaller I/O, so the overhead of I/O is smaller in the overall process, and

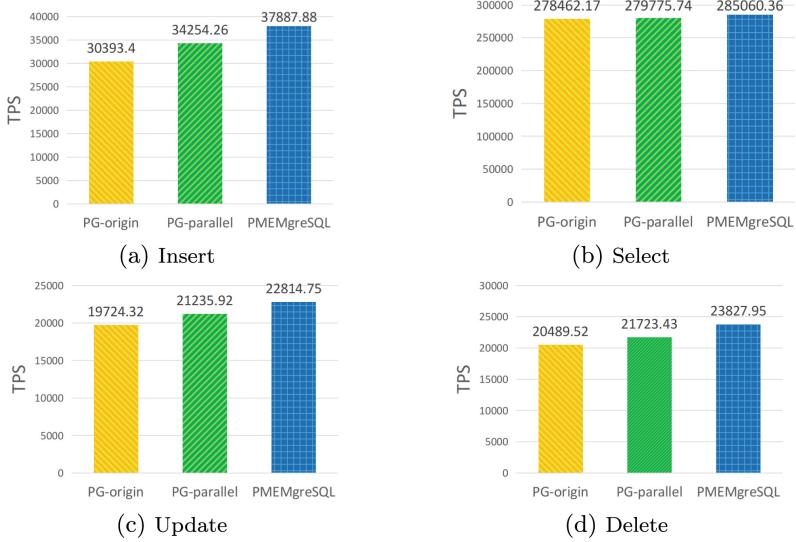


Fig. 3. Experiment Results for Parallelism Control

thus the benefits gained from concurrency control are less significant. This is reflected in the 16.3% and 9.7% performance improvements of PMEMgreSQL compared with PG-origin and PG-parallel.

From the performance comparison of the four basic operations, our PMem-aware parallel control system shows a pronounced performance improvement. The results validate our design and demonstrate the potential of PMem technology in optimizing database systems for modern, data-intensive applications.

3.3 Interleave Logging Experiment

To assess the effectiveness of our cross-NUMA logging algorithm, termed ‘Interleave’ in this experiment, we leverage PMem on two NUMA nodes to perform a comparative analysis on Insert, Update, and Delete operations. The Select operation is not included due to its lack of involvement in WAL disk writing. Since the original PG only allows single WAL space, we use the motherboard’s interleaving feature to combine two PMem devices within the same NUMA node into a single logical device, and pin the PG processes to one of the NUMA nodes to achieve inner-NUMA or cross-NUMA PMem writes, creating two comparative setups: ‘Inner-NUMA’ and ‘Cross-NUMA’. We also set the original version of PG without pinning as control group, named ‘Origin’.

As shown in Fig. 4, original PG has the lowest performance, even lower than Cross-NUMA versions. We believe it is due to cache-coherence, which incurs additional overhead when multiple cores access data from different NUMA nodes concurrently. While Cross-NUMA utilizes remote NUMA, less consistency overhead incurs. Inner-NUMA, due to accessing local PMem, performs better than

Cross-NUMA by omitting the cross-NUMA communication. On the other hand, our Interleave approach ensures that threads access their local NUMA, differing from Inner-NUMA in that it has more CPUs and DRAM available, thus delivering better performance under high load conditions.

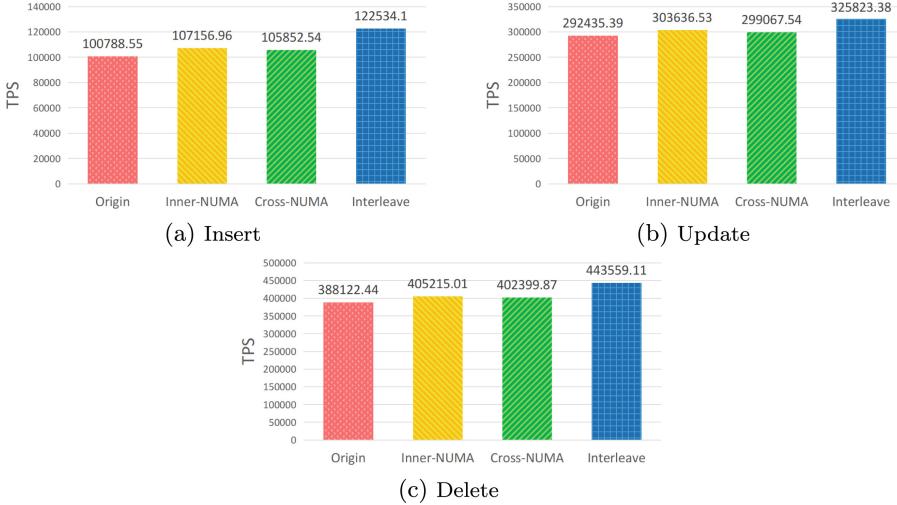


Fig. 4. Experiment Results for Interleave Logging

Performances of the Insert operation show that Interleave improves by 21.6% compared with the original PG, and by 14.4% and 15.8% compared with Inner-NUMA and Cross-NUMA, respectively. The enhancement is quite significant. While the Update operation, involving only a part of the Write operations, shows a relatively limited performance improvement in the experiments, with a 11.4% increase compared with the original PG. When compared with Inner/Cross-NUMA, it exhibits a similar level of enhancement, namely 7.3% and 9.0% respectively. The Write operation ratio in the Delete operation is closer to that of the Insert operation. Due to the presence of more writes, our algorithm also shows a significant performance improvement in the Delete operation, with optimization effects of 14.3%, 9.5%, and 10.2% when compared with the original PG, Inner-NUMA, and Cross-NUMA respectively.

From the experiment results above, it is evident that compared with the original PG's disk writing strategy, when we actively pin PG processes to specific NUMA nodes, whether local or remote, there is a stable improvement in disk writing performance. We attribute this finding to the importance of cache-coherence for PG performance under high load conditions. Nevertheless, due to the inherently limited bandwidth provided by a single PMem device, distributing log buffers across PMem devices in different NUMA nodes, allowing for concurrent log writing successfully balance the loss of accessing remote NUMA and

the loss of wait time for limited bandwidth, which is approved by the noticeable performance boost of our Interleave strategy.

4 Related Work

The literature on PMem reveals many innovations that aim to enhance database I/O performance. Tair-PMem, presented by Gong C et al., is a Non-Volatile Memory Database that capitalizes on PMem's consistency and high throughput [2]. Wu K et al.'s NyxCache demonstrates the efficiency of multi-tenant persistent memory caching, which is crucial for optimizing file and storage system latency [5]. Xu J and Swanson S's NOVA file system and Zhu B et al.'s Octopus file system both showcase how PMem can be utilized to reduce latency and increase throughput in storage systems [6, 12]. Chen Y et al.'s FlatStore and Liu J et al.'s LB+-trees highlight advancements in key-value storage and indexing for PMem, respectively [1, 3]. Neal I et al.'s work on rethinking file mapping and Yang J et al.'s empirical guide on scalable PMem provide valuable insights for optimizing database systems for PMem [4, 8]. Zhou D et al.'s ODINFS and Xu Q et al.'s work on I/O Transit Caching further the understanding of scaling PMem performance [7, 11]. Ye C et al.'s approach to enabling atomic durability with transiently persistent CPU cache is a significant contribution to data consistency in PMem-based systems [9]. Zheng S et al.'s TPFS is a file system designed to exploit the unique characteristics of heterogeneous memory architectures, addressing the challenges traditional file systems face with managing data across different memory tiers in the era of non-volatile memory technologies like PMem [10].

These studies in data consistency and high concurrency scenarios collectively provide a foundation for our work on PMEMgreSQL, focusing on the challenges of concurrent processing and NUMA architectures within the context of PG.

5 Conclusion

In this paper, we present PMEMgreSQL, a high-performance adaptation of PG that is specifically optimized for persistent memory. This system is designed to tackle the challenges associated with concurrent processing and the complexities introduced by NUMA architectures. In summary, PMEMgreSQL is an innovative approach to database management that harnesses the power of persistent memory while addressing the key challenges of concurrent processing and NUMA-related performance issues. By implementing a PMem-aware parallel control mechanism and a cross-NUMA logging interleave strategy, PMEMgreSQL aims to provide an efficient database system capable of meeting the demands of modern data-intensive applications.

References

1. Chen, Y., Lu, Y., Yang, F., Wang, Q., Wang, Y., Shu, J.: Flatstore: an efficient log-structured key-value storage engine for persistent memory. In: ASPLOS '20: Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, 16–20 March 2020, pp. 1077–1091 (2020)
2. Gong, C., et al.: Tair-PMem: a fully durable non-volatile memory database. Proc. VLDB Endow. **15**(12), 3346–3358 (2022)
3. Liu, J., Chen, S., Wang, L.: LB+-trees: optimizing persistent index performance on 3dpoint memory. Proc. VLDB Endow. **13**(7), 1078–1090 (2020)
4. Neal, I., Zuo, G., Shipley, E., Khan, T.A., Kwon, Y., Peter, S., Kasikci, B.: Rethinking file mapping for persistent memory. In: 19th USENIX Conference on File and Storage Technologies, FAST 2021, 23–25 February 2021, pp. 97–111 (2021)
5. Wu, K., et al.: NyxCache: flexible and efficient multi-tenant persistent memory caching. In: 20th USENIX Conference on File and Storage Technologies, FAST 2022, Santa Clara, CA, USA, 22–24 February 2022, pp. 1–16 (2022)
6. Xu, J., Swanson, S.: NOVA: a log-structured file system for hybrid volatile/non-volatile main memories. In: Brown, A.D., Popovici, F.I. (eds.) 14th USENIX Conference on File and Storage Technologies, FAST 2016, Santa Clara, CA, USA, 22–25 February 2016, pp. 323–338 (2016)
7. Xu, Q., Jiang, Q., Wang, C.: I/O transit caching for PMem-based block device. CoRR abs/2403.06120 (2024). <https://doi.org/10.48550/ARXIV.2403.06120>
8. Yang, J., Kim, J., Hoseinzadeh, M., Izraelevitz, J., Swanson, S.: An empirical guide to the behavior and use of scalable persistent memory. In: 18th USENIX Conference on File and Storage Technologies, FAST 2020, Santa Clara, CA, USA, 24–27 February 2020, pp. 169–182 (2020)
9. Ye, C., Chen, M., Jiang, Q., Wang, C.: Enabling atomic durability for persistent memory with transiently persistent CPU cache. CoRR abs/2210.17377 (2022). <https://doi.org/10.48550/ARXIV.2210.17377>
10. Zheng, S., Hoseinzadeh, M., Swanson, S., Huang, L.: TPFS: a high-performance tiered file system for persistent memories and disks. ACM Trans. Storage **19**(2), 20:1–20:28 (2023). <https://doi.org/10.1145/3580280>
11. Zhou, D., Qian, Y., Gupta, V., Yang, Z., Min, C., Kashyap, S.: ODINFS: scaling PM performance with opportunistic delegation. In: 16th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2022, Carlsbad, CA, USA, 11–13 July 2022, pp. 179–193 (2022)
12. Zhu, B., Chen, Y., Wang, Q., Lu, Y., Shu, J.: Octopus⁺: an RDMA-enabled distributed persistent memory file system. ACM Trans. Storage **17**(3), 19:1–19:25 (2021). <https://doi.org/10.1145/3448418>



The Development of a TLA⁺ Verified Correctness Raft Consensus Protocol

Hua Guo^{1,2(✉)}, Yunhong Ji², and Xuan Zhou³

¹ Scuptio, No. 988 Zhongcun Road, Shanghai 201109, China
guohua@scupt.com

² Renmin University of China, No. 59 Zhongguancun Street, Beijing 100872, China
{guohua2016, jiyunhong}@ruc.edu.cn

³ East China Normal University, No. 3663 North Zhongshan Road, Shanghai 200062, China
xzhou@dase.ecnu.edu.cn

Abstract. Distributed consensus protocols require significant effort to design, develop, and verify correctly. Traditional software quality assurance methods rely on developers creating extensive test cases to cover all code branches, which depend heavily on human experience and prolonged testing. While formal methods offer a reliable means of designing correct protocols, they ensure correctness only at the design level. Ensuring implementation correctness remains challenging once it deviates from its original design. This paper introduces our development of a verified correctness Raft protocol using an innovative specification-driven approach. We first specified Raft using TLA⁺ and verified its correctness with a model checker. Subsequently, we implemented the protocol based on this verified specification. Finally, we employed model checker tools to automatically generate test cases covering the entire design space for implementation verification, ensuring the implementation is an exact specification refinement. This approach ensures the correctness of both the implementation and the design.

Keywords: Raft · TLA⁺ · Distributed System

1 Introduction

Consensus algorithms are fundamental protocols for building highly available distributed systems. Paxos [9] and Raft [13] are prominent examples. Historically, consensus protocols have been regarded as complex and challenging to comprehend. Developing, testing, and ensuring their correctness presents significant difficulties. Traditional software quality assurance methods typically rely on experts creating exhaustive test cases to cover all code branches. This approach is highly dependent on the expertise and diligence of individuals, making it prone to errors and labor-intensive. Consensus protocols function within asynchronous networks, where the non-determinism introduced by asynchronous communication complicates the manual writing of test cases to ensure correctness. The presence of long-standing, undiscovered flaws in many widely used systems' consensus protocols

underscores the inherent difficulty of this task. Recently, some state-of-the-art approaches [6, 11, 17] have employed formal methods to generate test cases for testing distributed systems. Inspired by this work, we have utilized a formal method-driven approach to develop a verified correct Raft consensus protocol. We provide a specification-driven developed Raft(SDRaft) implementation, an open-source¹, TLA⁺ verified, fully functional Raft protocol that includes leader election, replication, member management, and log compaction. Scuptio, a startup software company, leverages SDRaft to construct highly available distributed filesystems.

Our contributions are as follows: We introduce a scalable, feasible specification-driven(SD) software quality assurance method applicable to consensus protocols and any other distributed systems. This paper is organized as follows: Sect. 2 discusses the related work. Section 3 presents using TLA⁺ to specify the Raft protocol, address state explosion, and verify correctness. Section 4 introduces the reference TLA⁺ action model and the validation of our Raft implementation using traces generated from the model. Section 5 evaluates the engineering cost to achieve verified correctness. Section 6 provides the conclusion.

2 Related Work

2.1 Raft Consensus Protocol

Raft [13] is a consensus algorithm to manage a replicated log in a distributed system. It was developed to be more understandable compared to other consensus algorithms like Paxos [9]. The primary goal of Raft is to ensure that distributed systems can achieve agreement on state changes even in the presence of faults.

2.2 Model-Based Testing

Model-based Testing [2, 15, 17] approaches use formal model to generate test cases to guarantee system quality. MongoDB used TLA⁺ to model and verify systems to ensure conformance between a specification and its implementation [15]. Kayfabe [2] can explore the entire state space of the model in testing and check each step of the program’s state. Mocket [17] utilized the state space generated through formal model checking to guide the testing of the system implementation and reveal any flaws present in the specific distributed system. This approach bridges the specification and implementation in a distributed system.

2.3 Formal Verification Frameworks

Formal verification frameworks(FVF) [4, 18] use mathematical methods to prove or disprove the correctness of a system’s design relative to its specifications. IronFleet [4] presents an approach for building high-assurance distributed systems by using a Dafny [10] framework. It introduces a programming methodology

¹ <https://github.com/scuptio/scupt-raft>.

Table 1. Specify non-deterministic action.

Action	Factor	Description
ReceiveMessage	external input	Node receives an incoming message
SendMessage	output	Node sends a message
Initialize	external input	Initialize the states of a node
RequestVote	timing	Trigger a timeout of the follower node and issue RequestVote message to other nodes
AppendLog	timing	Trigger a timeout of the leader node and issue AppendLog message to other nodes
ClinetRequest	external input	The client request append a value to leader
Restart	failure	The node restart
ReConfiguration	external input	Change the member setting of the cluster
LogCompaction	external input	Compact logs to reduce storage usage

and toolkit that combines formal verification and practical implementation techniques to ensure the correctness and robustness of distributed systems. Verdi [18] is a Coq [1] framework for building formally verified distributed systems. Verdi invokes Coq to translate its verified code into OCaml.

3 Specify Raft Consensus Protocol and Verify Correctness

The key components and operations of the Raft consensus algorithm include leader Election, log replication, log compaction, and membership Changes. We specify Raft using TLA⁺ and I/O automata [12] abstraction.

3.1 I/O Automata Abstraction

I/O automata are used to model distributed systems, focusing on interactions between components (input/output/internal actions) and their state transitions. We use a message passing I/O automata model to formal Raft. We map Raft's concepts (nodes, terms, log entries, elections, etc.) to I/O automata components. 1) States: Represent the state of each node, including current term, log entries, commit index, etc.; 2) Actions: Represent Raft events like receiving a vote, log replication request, election timeout, etc.; 3) State Transitions: Define how actions affect states (e.g., how a vote request changes a candidate to a leader); 4) Task Partition: Partition the I/O automata task by server node; each node was assigned a unique node ID.

3.2 Specify Non-deterministic Behavior

The challenge in developing distributed systems lies in their non-deterministic behavior. Deterministic behavior in software refers to the characteristic where a

program or system consistently produces the same output given the same input. This behavior means that the execution of the software follows a predictable and repeatable sequence of operations, leading to identical results every time it runs under the same conditions. For instance, a sorting algorithm like Quicksort [5], when given a specific input array, always sorts the array into the same order, demonstrating deterministic behavior. Deterministic behavior simplifies the testing and debugging of software since issues can be reliably reproduced and isolated.

In contrast, asynchronous network communication, where message delivery can be influenced by various factors such as network traffic, resulting in variable timing and order of message reception, exemplifies non-deterministic behavior [12]. Concurrency, failures, external inputs, timing, and randomization are the primary factors leading to non-deterministic behaviors. Distributed systems that process messages concurrently over asynchronous networks, such as those using the Raft consensus protocol, exhibit many non-deterministic behaviors. The non-deterministic behaviors of the Raft consensus protocol are enumerated in Table 1. We use an asynchronous message-passing style and maintain a variable to track messages sent within the system. The duplication, loss, and out-of-order delivery of messages are simulated. Node failures are not specified directly in the specification; instead, we consider a node failure as a period of inactivity followed by a *Restart* action. More details about the specification and the non-deterministic actions can be found in the TLA+ files in the source code repository.

3.3 Mitigate State Explosion of the Model

State explosion refers to the rapid growth of the number of states in a system’s state space as the system’s complexity increases. This phenomenon could be problematic in model checking, where the goal is to systematically explore all possible states of a system to verify its properties. As the model grows, the number of states can grow exponentially, making exploring all states within reasonable time and memory limits infeasible.

To mitigate state explosion, we employ several strategies. First is symmetry and view reduction [8]. By identifying symmetrical behaviors and treating them as equivalent, we can reduce the number of distinct states that need exploration. The view reduction technique ignores non-trivial states that do not significantly impact the verification. Second, we use bounded model checking. We limit the depth of the search to a certain number of steps, focusing on finding counterexamples within this bound. This approach helps manage the state space by constraining the exploration to a feasible subset. Third, we use a staged approach. We begin with a small model to explore the valid state space. Once the valid space is identified, it is dumped and used as the initial state for a new model-checking procedure focusing on a specific action. This process is repeated until the entire state space is explored. These strategies collectively help manage the state space more effectively, making it feasible to verify complex systems despite the potential for exponential growth in the number of states.

3.4 Model Check and Verify Correctness

Raft's correctness contains safety and liveness properties. Safety properties include leader election safety and log consistency. Leader election Safety ensures that at most one leader is elected in a given term. Consistency guarantees that if two logs contain an entry with the same index and term, then the logs are identical in all preceding entries. Liveness properties also encompass leader election and replication liveness. Leader Election liveness ensures that a leader is elected in each term. Log replication liveness guarantees that if a leader has decided to append an entry to its log, this entry will eventually be committed and visible to all servers. We executed the TLC model checker [8] on the Raft TLA⁺ specification and defined the correctness properties. If these properties were not violated, we can have confidence in confirming the correctness of this model. We use the profiling feature [8] provided by the TLA⁺ toolbox to ensure that all state spaces of the model are explored. While proofs using the TLA⁺ proof system may offer more accurate representations, we believe proofs are not essential for a well-designed algorithm such as Raft.

4 Validate Raft Consensus Implementation

4.1 Reference TLA⁺ Action Model

We use a reference TLA⁺ action model to map the action of specification and the program state and behaviors. This mechanism was provided by our development tool sedeve-kit [16]. This paper mainly focuses on the development of the Raft protocol. The development tool sedeve-kit is beyond the scope of this paper, and further information can be found in sedeve-kit [16]. In our Raft TLA⁺ specification, we use auxiliary variables *action* and *state* to keep each action for later use to validate the implementation. An example of the information kept by the *action* and *state* variable are shown in Fig. 1. Figure 1 shows the *RequestVote* action of a follower node after a timeout period and the state of the nodes after running this action. The action keeps the action type (Input), source and destination node (the same in this example, it simulates a timeout event), and the payload. The state keeps the essential member values of the state machines, which state has been changed by the action.

After running the TLC model checker, sedeve-kit can save the valid *action* and *state* values to the database and generate valid traces that can explore all state space of the model to validate the implementation to guarantee the system's behavior is the refinement of the specification.

We implemented the Raft consensus algorithm by mirroring our TLA⁺ specification. We validated our Raft implementation using deterministic testing facilitated by the testbed provided by the sedeve-kit. In this testing setup, the system is designed to receive incoming action messages from an overridden channel provided by the testbed. Each action is processed sequentially, without concurrency, thereby eliminating all sources of non-determinism. The testbed exhaustively explores all possible state compositions within the design space.

```

1  {
2      "action": {
3          "Input": {
4              "source": 3,
5              "dest": 3,
6              "payload": "RequestVote"
7          }
8      }
9 }

```

```

1  {
2      "state": {
3          "role": Candidate,
4          "current_term": 3,
5          "log": [],
6          "voted_for": 3
7      }
8 }

```

Fig. 1. RequestVote action and the state of the nodes after running this action.

```

1 fn handling_action_input(&self, msg:Message) {
2     // input! macro work only when deterministic testing
3     input!(msg);
4     ....
5 }

```

Fig. 2. The node handles an input message action.

Each test case is represented as a trace, a finite sequence of action and state pairs, the state corresponding to the results from those actions. Formally, a trace T is a sequence defined as:

$$a_1, s_1, a_2, s_2, \dots a_n, s_n$$

where a_i is the i th action of the trace, and s_i is the system's state after executing action a_i . The testbed processes each action a_i in sequence T . After executing action a_i , it verifies the system state by asserting that its current state matches the expected state s_i . If the system receives an action a_i , then it yields an action a'_{i+1} that does not match the expected following action a_{i+1} , the testbed will trigger a timeout to report the inconsistency. Validation fails if the action sequence or state assertions are inconsistent with the expected trace of the model. Due to the deterministic nature of this testing approach, any inconsistencies can be reliably reproduced and debugged, allowing us to identify and explain the cause of any bugs.

Figure 2 illustrates the override of the message channel to handle an input action. During deterministic testing, the *input* macro translates to source code that communicates with the testbed to validate the trace's consistency. This *input* macro is omitted in the system's released version. Figure 3 demonstrates the assertion of the system state against the expected model state after processing an input action.

4.2 Encapsulate the Verified Deterministic Code

```

1 fn checking_after_input_action(&self, state:State) {
2     // equal on Leader, Follower, Candidate
3     assert_eq!(state.role, self.role);
4     // equal on the Raft log
5     assert_eq!(state.log, self.log);
6     // equal on voted for node ID
7     assert_eq!(state.voted_for, self.voted_for);
8     // equal on the current term
9     assert_eq!(state.current_term, self.current_term);
10    ....
11 }
```

Fig. 3. After handling an input action, the node checks its states to assert its state is correct.

```

1 fn follower_node_timeout(&self) {
2     if self.role == Follower {
3         // todo! wait on a period timeout ...
4         // send RequestVote RPCs to other nodes
5         self.request_vote();
6         ....
7     }
8 }
```

Fig. 4. The follower node sends RequestVote RPCs to other nodes.

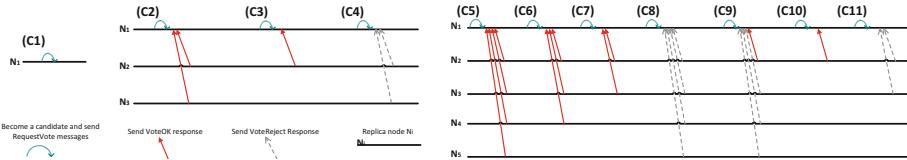
We encapsulated the verified deterministic code to handle real-world non-determinism. Figure 4 provides an example: after a timeout period, the follower node sends *RequestVote* RPCs to other nodes. Since we have verified and gained confidence in the correctness of the function *request_vote*, the additional naive code does not pose a significant trouble. Ultimately, the difference between our final release code and our verified, tested code lies in two parts: one is encapsulating code to handle real-world non-determinism, as illustrated in Fig. 4, and the other comes from the helper code for running deterministic testing, as shown in Fig. 2 and Fig. 3.

4.3 Conduct Validation Using Deterministic Testing

We validate our implementation using deterministic testing with our testbed and test cases. Once we successfully pass all test cases generated by our previously constructed model, we gain confidence that our implementation accurately reflects the specified behaviors. This ensures correctness in both the specification and the implementation within the design space.

Table 2. SDRaft and TiKV-Raft comparison.

Comparison Matrix	SDRaft	TiKV-Raft
Leader Election + Log Replication?	Y	Y
Persistence?	Y	Y
Membership Changes?	Y	Y
Log Compaction?	Y	Y
Programming Language	Rust	Rust
SLOC total	9925	20143
SLOC testing	2286	9328
SLOC TLA ⁺ specification	2797	X
SLOC test input data lines	X (auto-generated)	528
Number of total test cases	8205852	254

**Fig. 5.** Tests all cases that may happen in leader election. Subcase C_1 – C_{11} .

5 Evaluation

We evaluate the SD approach by considering its potential engineering costs and benefits to ensure system quality. First, we compare it with the traditional approach, which relies on expert-designed test cases. Next, we compare it with state-of-the-art work. By examining these comparisons, we aim to provide a comprehensive understanding of the impact and effectiveness of the SD approach in development.

5.1 Comparison With Traditional Approach

First, we compare SD testing against expert design testing. In our comparison, we utilized an open-source Raft implementation, raft-rs [14], used by TiKV [7], an open-source key-value database (referred to as TiKV-Raft later), as a baseline. TiKV-Raft is the Rust language implementation of Etcd-Raft [3], and the test cases of TiKV-Raft are also inherited from Etcd-Raft. SDRaft has the same functionalities as TiKV-Raft, as shown in Table 2. TiKV-Raft employs many test cases to evaluate Raft, and we meticulously gathered data on the Raft test case code and data in TiKV-Raft, which necessitated careful manual design. This design heavily relies on developers' understanding of the Raft protocol. SD testing approach generates these test cases and data using tools, thereby eliminating

Table 3. SDRaft and Verdi-raft comparison.

Comparison Matrix	SDRaft	Verdi-Raft
Leader Election + Log Replication?	Y	Y
Persistence?	Y	Y
Membership Changes?	Y	X
Log Compaction?	Y	X
Formal Language	TLA ⁺	Coq
Programming Language	Rust	OCaml
SLOC specification	2797	12550
Correctness Assurance	Model check	Proof
Relation Between Implementation and Specification	Refinement	Homogeneous

the need for manual design. Most TiKV-Raft test cases can be covered by test cases automatically generated by our SD approach. Figure 5 shows a test case example of TiKV Raft, which tests all cases that may happen in leader election during *VoteRequest* RPCs. Our test cases generated by the tool can cover all these cases.

We use a simple approach, SLOC (source lines of code, including comments but excluding empty lines), as a standard to estimate engineering costs. Table 2 shows the comparison TiKV-Raft and SDRaft. TiKV-Raft uses more testing code (more than 9000 SLOC) for its core functionality (10000+ SLOC). TiKV-Raft’s test data(additional 528 SLOC) must also be carefully maintained by the developer, which is challenging work. SDRaft also generates a significantly more significant number of test cases with only a tiny amount of formalized specification code lines, and it covers a higher proportion of the code compared to TiKV-Raft. Most importantly, the SD approach relies on something other than experts to design the test cases and prepare test case data, which can be tedious, laborious, and error-prone.

5.2 Comparison With State of the Art Work

Verdi-Raft is a Raft implementation developed using a formal verification framework (FVF) [18]. Table 3 compares Verdi-Raft and our SDRaft implementation. The Verdi-Raft implementation includes only the basic Raft algorithm, omitting advanced features such as log compaction and membership changes. In contrast, our SDRaft implementation supports full functionality. Additionally, Verdi-Raft and SDRaft utilize different formal languages, programming languages, and correctness assurance methods. Despite offering fewer functionalities, Verdi-Raft uses significantly more lines of specification code (SLOC) than our work. Applying FVF to Verdi-Raft requires substantially higher engineering effort than our SD approach. While FVF provides a rigorous method for proving system correctness, offering high assurance and comprehensive coverage, it comes with

increased complexity and resource demands. Our SD approach, on the other hand, offers a more practical and flexible method for validating system correctness against specifications. SD approach is easier to apply but may provide less comprehensive coverage and assurance.

6 Conclusion

We have presented a detailed approach to implementing and verifying the correctness of the Raft consensus protocol using a TLA⁺ specification-driven approach. By leveraging deterministic testing facilitated by the sedeve-kit testbed, we ensured that our Raft implementation was a refinement of its formal specifications. Through this work, we demonstrated that the specifications-driven approach could effectively validate system correctness against specifications.

References

1. The Coq Proof Assistant. <https://coq.inria.fr/>
2. Dorminey, S.: Kayfabe: model-based program testing with TLA+/TLC. In: TLA+ Conference 2020 (2020)
3. Etcd: etcd-raft. <https://github.com/etcd-io/raft.git/>
4. Hawblitzel, C., et al.: IronFleet: proving practical distributed systems correct. In: Proceedings of the 25th Symposium on Operating Systems Principles, SOSP 2015, Monterey, CA, USA, 4–7 October 2015, pp. 1–17. ACM (2015)
5. Hoare, C.A.R.: Quicksort. *Comput. J.* **5**(1), 10–15 (1962)
6. Hua Guo, X.Z.: Specification-driven development with TLA+. In: TLA+ Conference 2024 (2024)
7. Huang, D., et al.: TiDB: a raft-based HTAP database. *Proc. VLDB Endow.* **13**(12), 3072–3084 (2020)
8. Kuppe, M.A., Lamport, L., Ricketts, D.: The TLA+ toolbox. In: Proceedings Fifth Workshop on Formal Integrated Development Environment, F-IDE@FM 2019, Porto, Portugal, 7th October 2019, EPTCS, vol. 310, pp. 50–62 (2019)
9. Lamport, L.: The part-time parliament. *ACM Trans. Comput. Syst.* **16**(2), 133–169 (1998)
10. Leino, K.R.M.: Dafny: an automatic program verifier for functional correctness. In: Clarke, E.M., Voronkov, A. (eds.) LPAR 2010. LNCS (LNAI), vol. 6355, pp. 348–370. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17511-4_20
11. Lesani, M., Bell, C.J., Chlipala, A.: Chapar: certified causally consistent distributed key-value stores. In: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, 20–22 January 2016, pp. 357–370. ACM (2016)
12. Lynch, N.A., Tuttle, M.R.: Hierarchical correctness proofs for distributed algorithms. In: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, Vancouver, British Columbia, Canada, 10–12 August 1987, pp. 137–151. ACM (1987)
13. Ongaro, D., Ousterhout, J.K.: In search of an understandable consensus algorithm. In: USENIX ATC, pp. 305–319. USENIX Association (2014)

14. PingCAP: raft-rs. <https://github.com/tikv/raft-rs/>
15. Schvimer, J., Davis, A.J.J., Hirschhorn, M.: extreme modelling in practice. *Proc. VLDB Endow.* **13**(9), 1346–1358 (2020)
16. Scuptio: Specification-Driven Development Kit. <https://github.com/scuptio/sedeve-kit/>
17. Wang, D., Dou, W., Gao, Y., Wu, C., Wei, J., Huang, T.: Model checking guided testing for distributed systems. In: EuroSys 2023, Rome, Italy, 8–12 May 2023, pp. 127–143. ACM (2023)
18. Wilcox, J.R., et al.: Verdi: a framework for implementing and formally verifying distributed systems. In: SIGPLAN, pp. 357–368. ACM (2015)



Robust Multi-vehicle Routing with Communication Enhanced Multi-agent Reinforcement Learning for Last-Mile Logistics

Hai Wang^{1,2}, Shuai Wang^{1(✉)}, Shuai Wang¹, and Xiaolei Zhou³

¹ Southeast University, Nanjing, China

{hai,shuaiwang,shuaiwang-iot}@seu.edu.cn

² JD Logistic, Beijing, China

³ National University of Defense Technology, Changsha, China

zhouxiaolei@nudt.edu.cn

Abstract. The Vehicle Routing Problem (VRP) is crucial for optimizing logistics in applications such as express systems, industrial warehousing, and on-demand delivery. Last-mile logistics present unique challenges due to dynamic and uncertain pickup demands, requiring real-time routing adjustments and efficient management of delivery schedules. Existing heuristic-based methods rely heavily on manual rules and are inadequate for highly dynamic environments, while RL-based methods lack models for cooperative Problems. To address these issues, we propose the Communication Enhanced Multi-agent Reinforcement Learning (CEMRL) framework. CEMRL utilizes Context Encoding to unify environment features and local observations and employs a transformer-based communication enhancement module for efficient multi-agent communication. Our extensive experiments on a real-world dataset demonstrate that CEMRL significantly outperforms state-of-the-art baselines in travel distance and overdue rates, validating its effectiveness in complex logistics scenarios.

Keywords: Vehical Routing · Multi-agent Reinforcement Learning · Last-mile Logistics

1 Introduction

The Vehicle Routing Problem (VRP) is crucial for applications like express systems, industrial warehousing, and on-demand delivery [1], aiming to optimize routing plans for minimizing travel expenses while fulfilling demands. Unlike instant delivery, last-mile logistics do not have a one-to-one correspondence between pick-up and delivery at the same site. Last-mile logistics face unique challenges as couriers deliver packages from a central depot to various locations, contending with the dynamic and uncertain pickup demands caused by

the randomness of pickup times and locations. This complexity requires real-time routing adjustments and balancing fixed delivery schedules with new pickups, introducing significant variability. Developing advanced algorithms for cooperative scenarios can reduce costs and enhance efficiency, ultimately improving customer experience with timely and reliable delivery services.

Existing solutions to the multi-vehicle routing problem with soft time windows include heuristic and RL-based methods. Heuristic methods, like tabu-embedded simulated annealing [2] and adaptive large neighborhood search [3], iteratively approach an optimal solution but depend heavily on predefined rules and are slow in dynamic settings. RL-based methods use reward feedback and neural networks to tackle decision-making in VRPs [4], though they mainly address standard scenarios and lack cooperative pickup and delivery problem models, with some having limitations in exploration and optimization [5]. Our proposed CEMRL framework, in contrast, effectively manages uncertain pickup demands and enhances communication and cooperation among vehicles.

The primary opportunities in our research stem from two key aspects. First, the availability of rich real-world logistics data provides robust support for our models, enabling us to capture the complexities of actual operational environments and receive timely feedback. This data-driven approach ensures that our solutions are grounded in real scenarios, enhancing their applicability and effectiveness. Second, the use of attention networks allows us to express multi-agent correlated features efficiently. This capability is crucial for capturing the intricate interactions between agents, thereby improving the overall decision-making process in dynamic and complex logistics environments.

However, our research also faces two challenges. The first challenge arises from the inherent uncertainty in pickup demands, which increases the complexity of the problem. This uncertainty necessitates real-time dynamic planning, which in turn amplifies the communication burden among agents as they constantly need to exchange information to adapt to changing conditions. The second challenge is related to scalability. As the scale of pickup and delivery operations increases, the effective exchange of information among agents becomes more difficult. Ensuring that each agent can access and process relevant information in a timely manner is critical for maintaining system performance and efficiency, especially in large-scale operations.

To address the aforementioned challenges, we propose the Communication Enhanced Multi-agent Reinforcement Learning (CEMRL) framework, tailored for vehicle routing under uncertain pickup requests in last-mile Logistics. This framework leverages Context Encoding to encode environmental features and agents' local observations into a unified representation, facilitating more effective decision-making across the system. Additionally, we designed a Communication Enhancement Module that reconstructs the multi-agent reinforcement learning communication mechanism using a Transformer-based architecture. This allows agents to selectively focus on the most relevant communication information, thereby achieving efficient multi-agent communication. Finally, we employ the Advantage Actor-Critic (A2C) method and train the framework based on the

Centralized Training with Decentralized Execution (CTDE) architecture, ensuring robust learning and performance in dynamic and complex logistics environments.

In particular, our main contributions are as follows:

- We are the first to propose the multi-vehicle routing problem in last-mile logistics under scenarios of uncertain pickup demands and soft time windows.
- We design a novel CEMRL framework that integrates context encoding and a transformer-based communication enhancement module to handle complex logistics environments. This module effectively addresses the challenges of increased communication burdens and scalability by enabling agents to selectively focus on the most relevant communication information, ensuring efficient multi-agent communication and coordination.
- We conducted extensive experiments on a real-world dataset over two months, comprising more than 432,000 delivery requests and 135,000 pickup requests. The experimental results demonstrate that our proposed algorithm outperforms state-of-the-art baselines in terms of travel distance, pickup, and delivery overdue rates, achieving optimal performance across all three metrics.

2 Preliminaries

2.1 Partially Observable Markov Game

This paper addresses a type of Markov Game [6] called a Partially Observable Markov Game (POMG), essential for modeling scenarios with multiple agents interacting in environments with incomplete information. In a POMG, the environment consists of global states S and each agent has private observations O_1, O_2, \dots, O_N , with $o_i \in O_i$ representing the partial information an agent perceives. Agents have possible actions A_1, A_2, \dots, A_N . The transition function T maps the current state and joint actions to a new state, reflecting the probabilistic nature of state transitions. The reward function r_i gives feedback to each agent based on the current state and action, representing immediate benefits or costs. The observation function o_i maps the global state to local observations, modeling partial observability. The initial state is defined by a distribution ρ . Each agent aims to maximize its cumulative discounted reward, given by $R_i = \sum_{t=0}^T \gamma^t r_i^t$, where γ is the discount factor determining the present value of future rewards.

2.2 Problem Formulation

To address the vehicle routing problem in last-mile logistics, we need to add soft time windows and constraints for delivery and pickup tasks to the model. The set of nodes is defined as $V = \{v_0, v_1, \dots, v_{M+1}\}$, where v_0 and v_{M+1} are the starting and returning depots, respectively. Node v_j represents a delivery or pickup point, where $1 \leq j \leq M$. Delivery demand is fixed within a delivery cycle, while pickup demand is dynamic.

The vehicle fleet consists of I vehicles, each with a capacity C_i . We introduce the variable $z_{jk}^i \in \{0, 1\}$, which indicates whether vehicle i travels from node v_j to node v_k . The variable T_j^i denotes the time vehicle i arrives at node v_j , and L_j^i represents the load of vehicle i at node v_j .

The objective is to minimize the total travel distance of all vehicles and the penalty for time windows:

$$\min \sum_{i \in I} \sum_{j \in V} \sum_{k \in V} e_{jk} z_{jk}^i + \sum_{i \in I} \sum_{j \in V} \text{penalty}(T_j^i), \quad (1)$$

$$\text{s.t. } \sum_{i \in I} \sum_{k \in V} z_{jk}^i = 1, \quad \forall j \in V \text{ (Pick-up & Delivery Nodes)}, \quad (2)$$

$$0 \leq L_j^i \leq C_i, \quad \forall j \in V, i \in I, \quad (3)$$

$$a_j \leq T_j^i \leq b_j + \text{penalty}(T_j^i), \quad \forall j \in V, i \in I, \quad (4)$$

$$\sum_{k \in V} z_{jk}^i = \sum_{k \in V} z_{kj}^i, \quad \forall j \in V, i \in I \quad (5)$$

where e_{jk} represents the distance between nodes v_j and v_k . Constraint (2) guarantees that each node v_j is visited exactly once by a vehicle. Here, z_{jk}^i indicates whether vehicle i travels from node v_j to node v_k , ensuring each node is visited precisely once. Constraint (3) ensures the load of each vehicle cannot exceed its capacity. Vehicles must complete tasks within the allowed time windows. For soft time windows, we introduce a penalty term $\text{penalty}(T_j^i)$ to represent the penalty for arriving outside the time window. Equation (5) ensures the continuity of vehicle routes.

3 Design of CEMRL

3.1 System Overview

We have designed a Context Encoding Module to process and integrate environmental and agent-specific features, transforming them into a unified representation for decision-making. Additionally, our Communication Enhancement Module facilitates efficient real-time information exchange among agents, improving coordination and mitigating partial observability issues. Together, these modules optimize routing and task allocation, ensuring efficient and cost-effective operations in dynamic environments. The system framework is shown in Fig. 1.

3.2 Context Encoding

In the context of last-mile logistics, our agents are represented by vehicles, with their state information including current parcel pickup and delivery counts, vehicle positions, remaining capacity, and the spatiotemporal details of the parcels. The agents' actions consist of selecting a vehicle to pick up or deliver at designated locations, with the final path being the shortest route from the vehicle's

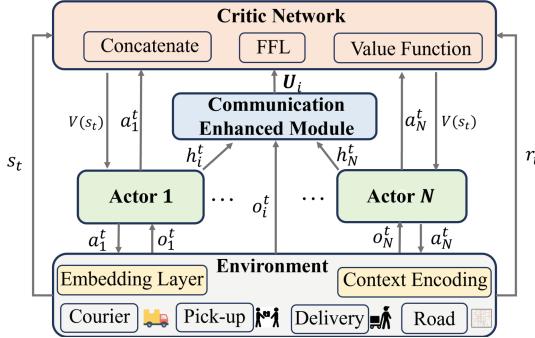


Fig. 1. Framework of CEMRL.

current position to the pickup or delivery spots. We integrate multiple vehicles and tasks to formulate the complete path for each vehicle, optimizing the entire delivery network.

In a multi-agent reinforcement learning environment, the designed context encoding module primarily processes environmental and agent feature information. First, for each agent's input features, we generate query (Q), key (K), and value (V) vectors through linear transformations. This step can be expressed as: $Q = W_Q x, K = W_K x, V = W_V x$, where W_Q, W_K , and W_V are the weight matrices for the linear transformations, and x is the input feature.

Next, we calculate the dot product of the queries and keys, and scale the result to generate the attention scores. This process can be simplified as: $\text{Attention} = \text{softmax}(QK^T / \sqrt{d_k})V$, where d_k is the dimension of the key vectors. The softmax function is used to normalize the attention scores, resulting in weighted values.

Then, we concatenate the outputs of the multi-head attention mechanism and apply a linear transformation to obtain the integrated feature representation: $\text{Concat_Linear} = W_O \cdot \text{Concat}(\text{head}_1, \dots, \text{head}_h)$, where head_i represents the output of each attention head, and W_O is the weight matrix for the linear transformation.

After that, we perform addition and normalization operations on the integrated features and the original input:

$$\text{Add_Norm_1} = \text{LayerNorm}(x + \text{Concat_Linear}) \quad (6)$$

Then, the features are further processed by a feedforward neural network, followed by another round of addition and normalization: $\text{Add_Norm_2} = \text{LayerNorm}(\text{Add_Norm_1} + \text{FeedForward})$, where FeedForward is the output of the feedforward neural network.

Finally, the output representation of the Context Encoding module is h_i , expressed as $h_i = \text{Add_Norm_2}$. Through these steps, the Context Encoding module effectively processes environmental and agent feature information, pro-

viding rich contextual representations for subsequent reinforcement learning decision-making processes.

3.3 Communication Enhanced Module

In the communication enhancement module for multi-agent reinforcement learning, we combine agent observations and internal states with Transformer technology to manage historical communication vectors. The process begins by embedding each agent's observations o_i and internal states h_i , which are then concatenated to form the communication vector c_i^t :

$$c_i^t = \text{Concat}(o_i W_o + b_o, h_i W_h + b_h) \quad (7)$$

Historical communication vectors are embedded and time-encoded, forming x_i^k , which undergoes multi-head self-attention within a Transformer model. The self-attention mechanism and the computations within are key to capturing the temporal dynamics and relationships among the agents:

$$\text{MultiHead}(X_i) = \text{Concat}(\text{Head}_1, \dots, \text{Head}_h) W^O \quad (8)$$

$$\text{Head}_j = \text{Attention}(X_i W_j^Q, X_i W_j^K, X_i W_j^V) \quad (9)$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_i}}\right) V \quad (10)$$

Following attention, residual connections and layer normalization are applied to stabilize and enhance the learning process:

$$Z_i = \text{LayerNorm}(X_i + \text{MultiHead}(X_i)) \quad (11)$$

The output from this layer feeds into a feed-forward network, undergoing another normalization to finalize the communication vector U_i , which is crucial for decision-making enhancements in the Actor and Critic networks:

$$U_i = \text{LayerNorm}(Z_i + \text{ReLU}(Z_i W_1 + b_1) W_2 + b_2) \quad (12)$$

This method effectively synthesizes observation data and internal dynamics using sophisticated Transformer architectures, optimizing the performance and decision-making capabilities within a multi-agent reinforcement learning system.

3.4 Model Training

We developed a cooperative Actor-Critic (A2C) framework [7] based on a Centralized Training with Decentralized Execution (CTDE) approach. This system includes a centralized critic network, $V_\omega^\pi(s, U)$, which uses policy outputs $\pi_{\theta_i, \phi}(v(i, t))$ and context embeddings h_i from each agent. These are integrated into a weighted sum, represented by the equation:

$$v_c = \text{LinearProjection}\left(\sum_{i=1}^N w_i \cdot h_i + U_i\right) \quad (13)$$

This vector v_c is then processed through dense layers to produce the state value estimate.

Training involves computing the advantage function $A^\pi(s, a)$ to determine the quality of actions compared to the average:

$$A^\pi(s, a) = r(s, a) + V^\pi(s', U'; w) - V^\pi(s, U; w) \quad (14)$$

This function is pivotal for updating the actor network parameters θ_i using policy gradient loss:

$$\nabla \mathcal{L}(\theta_i) = \mathbb{E} [\nabla_{\theta_i} \log \pi_{\theta_i, \phi}(a|o_i, U_i) A^\pi(s, a)] \quad (15)$$

Similarly, shared parameters ϕ are updated to enhance actor networks' components, and the critic network parameters ω are refined to improve the accuracy of value estimates:

$$\mathcal{L}(\omega) = \mathbb{E} [(A^\pi(s, a))^2] \quad (16)$$

This framework enables effective cooperative learning through a centralized critic, improving stabilization and learning efficacy, while allowing for decentralized execution based on individual observations and communication vectors.

4 Evaluation

4.1 Experimental Setup

Dataset and Evaluation Configuration. We used a real-world dataset from Jing-Dong Logistics featuring over 432,000 delivery and 135,000 pickup requests from a 2-month period, detailing courier trajectories and parcel information in a city's central area. Our evaluation used dynamic route planning triggered by new pickup tasks, grouping locations within the same Area of Interest, and setting a daily variance in pickup requests to handle uncertainties effectively. Couriers' speeds were set at 25 km/h with specific service times for delivery and pickup based on historical data. Our model and baselines were implemented in Python using Pytorch on 3090Ti GPUs, optimizing with Adam at a learning rate of 1e-4 and specific penalty coefficients for delivery and pickup tasks.

Baselines. We evaluated our model against several methods: real courier routes (Experience, EXP), Ant Colony Optimization (ACO) [8], Tabu Search [9], and RL-based approaches including RL-VRP [10], MAAM [11], MDAM [12], and MAPDP [13].

Metrics. The metrics used were Traveling Distance (TD) for efficiency and Overdue Rate, including Pick-up Overdue Rate (POR) and Delivery Overdue Rate (DOR), to assess timeliness.

4.2 Overall Performance

To validate the effectiveness of our approach, we conducted extensive experiments on real-world data, and the results are presented in Table 1. The experimental results show that the heuristic-based methods EXP, ACO, and Tabu Search perform relatively poorly in terms of overdue rates and total travel distance, while the RL-based methods RL-VRP, MAAM, MDAM, and MAPDP significantly improve these metrics, with MAPDP being the best among the RL methods. However, our proposed CEMRL algorithm outperforms all other methods in all evaluation metrics, exhibiting the lowest overdue rates and the shortest total travel distance, demonstrating its high efficiency and reliability in handling multi-vehicle routing problems.

Table 1. Overall Performance. Bold scores are for the best values.

Method	POR	DOR	TD (km)
EXP	0.232	0.125	22.036
ACO	0.189 ± 0.021	0.133 ± 0.003	20.491 ± 1.026
Tabu Search	0.176 ± 0.010	0.105 ± 0.008	20.450 ± 1.176
RL-VRP	0.155 ± 0.006	0.092 ± 0.004	19.233 ± 0.847
MAAM	0.147 ± 0.012	0.078 ± 0.011	19.882 ± 0.911
MDAM	0.138 ± 0.006	0.079 ± 0.006	19.245 ± 1.267
MAPDP	0.121 ± 0.004	0.065 ± 0.003	18.985 ± 2.221
CEMRL	0.092 ± 0.008	0.041 ± 0.004	17.782 ± 0.503

4.3 Ablation Study

We assessed the impact of specific components in our CEMRL design by comparing it with two variants: CEMRL-EN, lacking the context encoding module, and CEMRL-CM, missing the communication enhancement module. Figure 2 and 3 show that CEMRL achieved the best performance, with a traveling distance of 17.782 km, significantly better than CEMRL-EN's 19.874 km and CEMRL-CM's 22.332 km. CEMRL also had lower overdue rates, particularly in pickups, underscoring the critical roles of context encoding and communication enhancement in improving outcomes.

4.4 Impact of Factors

The Impact of the Number of Agents. Figure 4 shows that increasing the number of agents significantly reduces the overdue rates for both pickup and delivery tasks. With 5 agents, the overdue rates for pickup and delivery tasks are around 0.20 and 0.15, respectively. With 25 agents, these rates drop to approximately 0.08 and 0.03, indicating improved system efficiency.

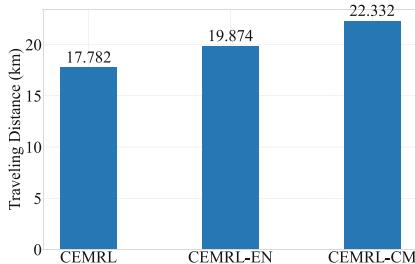


Fig. 2. The Effect of Ablation Study on Traveling Distance.

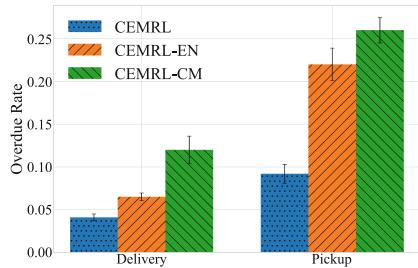


Fig. 3. The Effect of Ablation Study on Overdue Rate.

The Impact of the Penalty Factor. Figure 5 illustrates that as the pickup penalty factor increases, the overdue rate for pickup tasks decreases from 0.15 to 0.07. The overdue rate for delivery tasks remains relatively stable, slightly increasing to around 0.07 at a penalty factor of 5. This suggests that higher pickup penalty factors effectively reduce pickup overdue rates with minimal impact on delivery tasks.

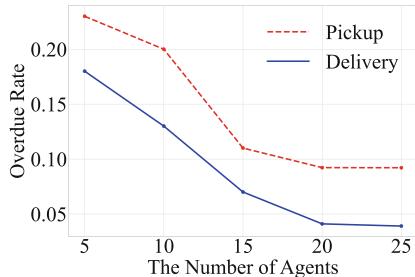


Fig. 4. The Impact of the Number of Agents.

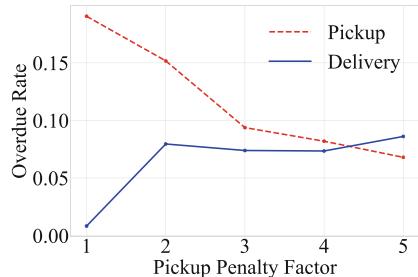


Fig. 5. The Impact of the Penalty Factor.

5 Related Work

Existing VRP solvers can be divided into two categories: heuristic-based methods and RL-based methods.

5.1 Heuristic-Based Methods

Heuristic-based solvers are usually iterative, aiming to eventually converge to the true optimum of the system [14, 15]. A tabu-embedded simulated annealing algorithm [2] has been proposed to solve large-scale VRP problems with time windows. An adaptive large neighborhood search heuristic method [3] has also

been proposed for solving VRP problems, incorporating regret insertion and six removal strategies. Although heuristic-based methods can produce near-optimal solutions in a more reasonable timeframe, they depend heavily on manually crafted rules and are significantly constrained by human expertise. Additionally, their online inference time remains inadequate when dealing with highly dynamic environments.

5.2 RL-Based Methods

Reinforcement Learning (RL) effectively solves decision-making problems using feedback rewards as training signals. Bello et al. [4] introduced an RL-based algorithm for combinatorial optimization, such as the Traveling Salesman Problem (TSP). Nazari et al. [10] used RNNs for capacitated VRP, and Kool et al. [16] improved it with an attention-based network. Xin et al. [12] developed a multi-decoder framework for fine-tuned solutions. Chen et al. [17] and Lu et al. [18] advanced RL techniques integrated with operations research for continual updates, although these methods generally focus on typical VRPs without supporting cooperative PDPs. Li et al. [5] developed frameworks for dynamic and single-vehicle PDPs, but gaps in cooperative PDP solutions persist. Zong et al. [13] introduced a multi-agent RL framework for VRPs and cooperative delivery, constrained by global communication assumptions. Yan et al. [19] employed conformal prediction with RL for routing, but struggled with the complexity of real-world demand randomness and lack of multi-agent modeling. In contrast, our proposed CEMRL addresses the multi-vehicle routing problem considering uncertain pickup demands and efficient communication and cooperation among multiple vehicles.

6 Conclusion

In this paper, we introduce the CEMRL framework to tackle the multi-vehicle routing problem in last-mile logistics with uncertain pickup demands and soft time windows. CEMRL integrates context encoding and a transformer-based communication module to improve agent coordination and decision-making in dynamic environments, enabling real-time data processing, effective communication, and adaptation to changing conditions, thus optimizing performance and efficiency in complex logistics operations. Experiments on a real-world dataset demonstrated that CEMRL significantly outperforms state-of-the-art baselines in minimizing travel distance and reducing overdue rates, proving its effectiveness in complex logistics scenarios.

References

1. Zong, Z., Feng, T., Xia, T., Jin, D., Li, Y.: Deep reinforcement learning for demand driven services in logistics and transportation systems: a survey. arXiv preprint [arXiv:2108.04462](https://arxiv.org/abs/2108.04462) (2021)

2. Ropke, S., Cordeau, J.-F.: Branch and cut and price for the pickup and delivery problem with time windows. *Transp. Sci.* **43**(3), 267–286 (2009)
3. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **40**(4), 455–472 (2006)
4. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. arXiv preprint [arXiv:1611.09940](https://arxiv.org/abs/1611.09940) (2016)
5. Li, J., Xin, L., Cao, Z., Lim, A., Song, W., Zhang, J.: Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **23**(3), 2306–2315 (2021)
6. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: 1994 Machine Learning Proceedings, pp. 157–163. Elsevier (1994)
7. Konda, V., Tsitsiklis, J.: Actor-critic algorithms. In: Advances in Neural Information Processing Systems, vol. 12 (1999)
8. Gambardella, L.M., Taillard, É., Agazzi, G.: MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows (1999)
9. Glover, F.: Tabu search—part I. *ORSA J. Comput.* **1**(3), 190–206 (1989)
10. Nazari, M., Oroojlooy, A., Snyder, L., zakáć, M.: Reinforcement learning for solving the vehicle routing problem. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
11. Zhang, K., He, F., Zhang, Z., Lin, X., Li, M.: Multi-vehicle routing problems with soft time windows: a multi-agent reinforcement learning approach. *Transp. Res. Part C: Emerg. Technol.* **121**, 102861 (2020)
12. Xin, L., Song, W., Cao, Z., Zhang, J.: Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 13, pp. 12 042–12 049 (2021)
13. Zong, Z., Zheng, M., Li, Y., Jin, D.: MapDP: cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 9, pp. 9980–9988 (2022)
14. Applegate, D., Bixby, R., Chvatal, V., Cook, W.: Concorde TSP solver (2006)
15. Helsgaun, K.: An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems. Roskilde: Roskilde University, vol. 12, pp. 966–980 (2017)
16. Kool, W., Van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint [arXiv:1803.08475](https://arxiv.org/abs/1803.08475) (2018)
17. Chen, X., Tian, Y.: Learning to perform local rewriting for combinatorial optimization. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
18. Lu, H., Zhang, X., Yang, S.: A learning-based iterative method for solving vehicle routing problems. In: International Conference on Learning Representations (2019)
19. Yan, H., Tan, H., Wang, H., Zhang, D., Yang, Y.: Robust route planning under uncertain pickup requests for last-mile delivery. In: 2024 Proceedings of the ACM on Web Conference, pp. 3022–3030 (2024)



A Dual-Tower Model for Station-Level Electric Vehicle Charging Demand Prediction

Qinyuan Li¹ , Lei Yao¹ , Shaolin Wang^{1,2}, Haoyang Che¹, and Yan Yi¹

¹ Zeekr Intelligent Technology, Hangzhou, China

¹ {qinyuan.li,Lei.Yao1,Shaolin.Wang1}@zeekrlife.com

² Geely Holding Group, Hangzhou, China

Abstract. With the rapid growth of electric vehicles (EVs), the efficient operation of EV charging stations has become crucial. Accurate demand prediction enables operators to respond effectively to demand fluctuations. This paper proposes a novel dual-tower architecture to predict demand changes caused by both price-related and price-unrelated factors. The left tower uses a Bi-GRU hybrid with 1D convolutional layers to predict baseline demand from price-unrelated factors. The right tower constructs a heterogeneous graph of stations to model price elasticity. An elasticity function then calculates the actual demand based on price elasticity and baseline charging. Our model employed meta-learning in the pre-training step to transfer knowledge from region-level to station-level data for enhancing accuracy. Experimenting on a real-world dataset from Zeekr Intelligent Technology with data from over 800 stations across China, our model significantly outperforms two comparison methods and could predict the charging demand effectively.

Keywords: Electric vehicle charging · Spatio-temporal prediction · Meta-Path · Meta-Learning

1 Introduction

Global sales of electric vehicles (EV) approached 14 million in 2023, representing 18% of all cars sold, an increase from 14% in 2022, accounting for more than one in five cars sold worldwide. According to world EV data [1], in the first quarter of 2024, electric car sales grew by around 25% compared to the same period in 2023. The market share of electric cars could reach up to 45% in China. The rapid growth in EV ownership has created a substantial market for electric vehicle charging stations (EVCS). However, EVCS operators face significant challenges. Charging demand varies widely between cities and areas, across different days and seasons. The increasing number of charging stations also brings competition, leading to unstable revenues. Therefore, effectively managing and optimizing charging station prices becomes crucial.

Dynamic pricing emerges as a potential solution to these challenges. It offers flexibility in adjusting prices based on competitive market conditions, enabling operators to maximize profits during peak demand and attract users during low demand, thereby enhancing overall profitability [2]. Price Elasticity of Demand (PED) represents the sensitivity of demand to price changes [3]. It aids in understanding the demand-price relationship, facilitating predictions of demand changes in response to price adjustments. Predicting precise charging demand with price adjustments is a key aspect of dynamic pricing implementation. However, in real-world scenarios, charging demand is influenced by various price-unrelated factors such as holidays, weather, and seasonal changes, as shown in Fig. 1. Therefore, considering both price-related and price-unrelated factors is essential when predicting charging demand.

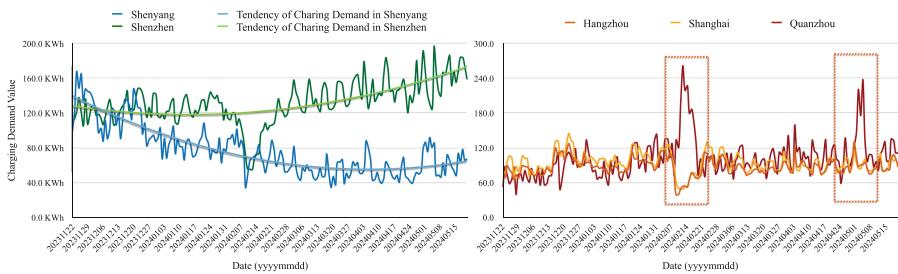


Fig. 1. The left figure shows that, as temperatures rise, the demand for electric vehicles gradually increases. The right figure demonstrates that charging demand peaks before travel seasons like weekends and holidays. During holidays, charging in developed cities with many non-local residents sharply decreases as people return to their hometowns.

Existing models often overlook the simultaneous consideration of price-related and price-unrelated factors affecting demand. To address this, we propose a dual-tower network by integrating both factors to improve charging demand predictions. Price-unrelated factors impact baseline charging demand, while price elasticity remains a valid measure of demand fluctuation sensitivity to price changes. Our model comprises two parts: the left tower predicts baseline charging volumes using a hybrid GRU neural network, and the right tower predicts price elasticity using a meta-path approach to model relationships among stations and price, with a customized elasticity coefficient equation to calculate the final charging demand prediction. Besides, we introduce a multi-step training method with meta-learning, utilizing region-level charging demand as pre-training data to extract time-series patterns and learn model parameters. Finally, validated on real data from over 800 charging stations operated by Zeekr, an EVCS operator and electric vehicle manufacturer, our model demonstrates the improvements in prediction accuracy of charging demand.

Our model offers several innovations in the field of EV charging prediction:

- **Dual-Tower Architecture:** Incorporates both price-related and price-unrelated factors.

- **Heterogeneous Map:** Considers how price-related factors affect charging demand on geographic-related stations.
- **Multi-Step Training Method:** Uses region-level data for pre-training to learn adaptable weights.
- **Verification with Real-World Data:** Proven effectiveness with real-world data from over 800 charging stations.

The subsequent sections provide a comprehensive analysis of our model, Sect. 2 reviews related work, Sect. 3 details our methodology, Sect. 4 presents model evaluation and comparison, and Sect. 5 summarizes findings and future improvements.

2 Related Work

Charging demand prediction is essential for dynamic pricing. Advances in machine learning and deep learning have led researchers to adopt sophisticated models for improved prediction accuracy. For instance, Yi et al. [4] combined travel chain analysis with a Monte Carlo search to predict charging demands, while Bao et al. [13] used CRF transition matrices and Learned Belief Propagation to model the stochastic nature of travel and charging behaviors, enabling adaptable probabilistic forecasts. RNNs, which can learn long-term dependencies in time series data, have been widely used in charging demand prediction. Wang et al. [5] utilized LSTM for short-term demand prediction, and Li et al. [11] considered various time steps to capture daily and weekly cycles. Spatial-temporal networks are also commonly used. GCNs learn spatial information, while GRU and LSTM extract temporal information [7–9]. Attention mechanisms have been introduced to extract spatial-temporal dependencies as well. Qu et al. [10] have employed attention mechanisms to enhance spatial and temporal dependency extraction, combining physical information meta-learning with graph attention networks (GAT) and improving model interpretability and flexibility.

Despite these advancements, existing models have limitations that our model aims to address. Most focus on region-level predictions, which differ significantly from station-level predictions due to varying demand at individual stations. They also often fail to account for both price-related and price-unrelated factors affecting demand. Furthermore, these methods infrequently do not generalize well due to limited training data. Our model integrates these factors to enhance predictive capabilities and generalizability.

3 Methodology

In this section, we introduce a dual-tower architecture designed to enhance EV charging demand prediction. The overall structure of the newly proposed dual-tower architecture is shown in Fig. 2. The Left Tower focuses on baseline charging prediction by employing a time series model and a meta-learning approach. The Right Tower is dedicated to price elastic coefficient prediction. A graphing

module is utilized to understand the price relationships among various charging stations. To integrate the predictions from both the left and right towers, a specialized PED equation is designed to combine the baseline predictions from the left tower with the price elasticity output from the right tower to calculate the overall charging demand.

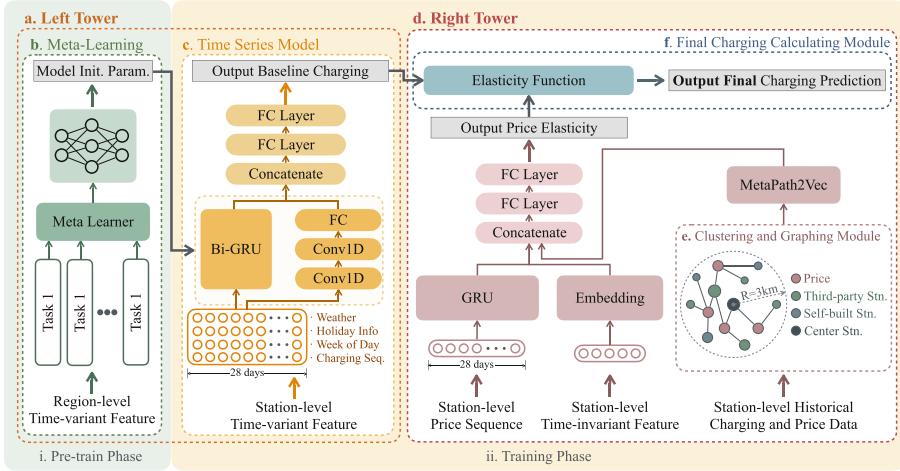


Fig. 2. Net-framework of our model.

3.1 Left Tower

The left tower aims to make the baseline charging prediction, deploying a Time Series Model and a Meta-learning phase as shown in Fig. 2 a. The Time Series Model employs a Bi-GRU (Bidirectional Gated Recurrent Unit) [6] to effectively capture temporal dependencies within the charging data. By leveraging meta-learning with region-level data for pre-training, the model's generalization capabilities are expected to be enhanced. This combination allows the left tower to produce accurate baseline predictions by learning from diverse regional patterns.

Time Series Model. The baseline charging demand prediction module focuses on price-unrelated factors influencing demand, analyzing user travel behavior and station characteristics. Station characteristics, including geographical location and type (fast, rapid, or standard charging), are treated as time-invariant factors considered in the right tower. In contrast, the left tower aggregates time-variant features to predict baseline charging demand.

Time-variant features model dynamic demand characteristics such as weather, day types, and holidays. Weather influences travel behavior, weekdays

and weekends exhibit distinct charging patterns due to commuting and recreational travel. Public holidays and weekends display unique travel and charging behaviors, affecting demand. These features capture the dynamism of charging trends over time, including seasonality and periodicity. As shown in Fig. 2 c, the Bi-GRU model is used to capture time-varying relationships in the data. Then, the output from the Bi-GRU layer is fed into one-dimensional (1D) convolutional layers. It aims to capture local features of the time series data, enhancing interpretability and performance in predicting charging demand. This approach provides a comprehensive prediction model to predict station charging from time-variant features.

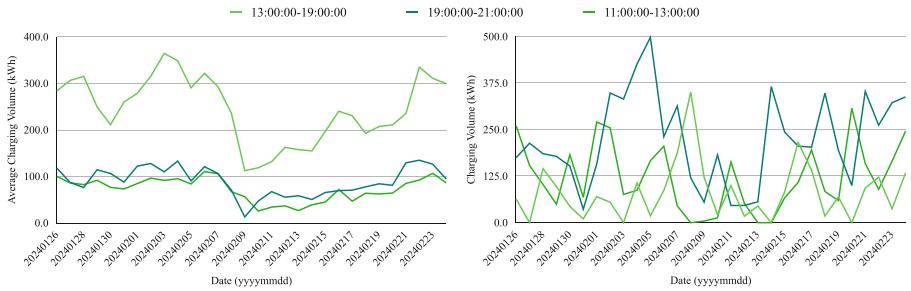


Fig. 3. Comparison of station-level and region-level demand fluctuations in 30 days for different time slots. The left figure shows the average charging volume of all stations and the right figure shows the charging volume of a single station.

Meta-Learning. In real-world scenarios, individual charging station demand varies significantly daily, while city-level aggregated data shows smoother trends, as shown in Fig. 3. This regularity makes meta-learning crucial.

Meta-learning [14] involves acquiring knowledge from various learning tasks to find model parameters that are globally optimal across all tasks. In our context, it captures regularities and dynamics in charging data from different levels of data sparseness. By integrating charging behaviors from multiple cities, meta-learning helps construct a robust model to effectively guide station-level demand prediction, addressing irregularities and improving accuracy. In our study, we establish a meta-learning model at the city level, dividing the data into support and query sets to train and identify parameters that can generalize well at the station level. By initializing parameters for station-level model training, we effectively transfer these regularities from city-level to station-level training.

3.2 Right Tower

Given the competitive nature of the charging market, studies indicate that price changes significantly influence consumer behavior. Our model, therefore, assumes

that charging demand is elastic to price, with the elasticity coefficient influenced by the price-related features of each station. By accurately assessing each station's price sensitivity, our model better reflects market dynamics and predicts the impact of pricing on charging demand.

Meta-Path. EVCS shows varying sensitivity to price changes, with nearby stations experiencing different demand levels even at the same price. Inspired by hotel demand prediction research [17], we use metapath2vec [16] to extract the relationship between prices and EVCS. We cluster self-built and third-party charging stations by geographic location to reflect competition and substitution dynamics, creating a heterogeneous graph from recent charging data. As shown in Fig. 2 e, the graph includes three types of nodes: prices (P), self-built stations (S_s), and third-party stations (S_t), with weights representing charging volumes.

We define three meta-paths: The meta-paths S_s-P-S_t and S_s-P-S_s connect charging stations through price nodes. The S_s-P-S_t path captures the substitutability and competition intensity among different charging stations at similar price levels, while S_s-P-S_s reflects the cooperation among stations from the same operator, helping to understand competition within the same price range. The $P_1-S_s-P_2$ path links price nodes through self-built charging stations, representing how price changes influence charging quantities. Stations with numerous price connections and less variation in connection weights indicate user insensitivity to prices, as demand remains stable despite price fluctuations. These paths help predict price elasticity and support more accurate charging demand forecasting by modeling the complex interactions between prices and charging stations.

Demand and Price Elasticity Function. In general, the effect of price on charging demand follows a downward-sloping demand curve [3]. To model this relationship accurately, we incorporate the elasticity coefficient σ_i^t , which measures the responsiveness of demand to price changes, as shown in Eq. (1).

$$Q_i^t = Q_i^{b,t} \times \left(\frac{P_i^t}{\bar{P}_i^t} \right)^{-\sigma_i^t} \quad (1)$$

The final charging output of the i_{th} station, Q_i^t , is determined by the predicted baseline demand $Q_i^{b,t}$, the input price P_i^t , the average price over the past 28 days \bar{P}_i^t , and the predicted price elasticity σ_i^t .

The baseline demand $Q_i^{b,t}$ is the output of the left tower, and the price elasticity σ_i^t is the output of the FC Layer from the right tower, where the left and right towers are closely linked. The elasticity coefficient function directly incorporates price changes by calculating the ratio of the input price to the average price, allowing demand to adjust accordingly. Besides, by ensuring the positive elasticity coefficient, our model maintains the inverse relationship, where higher prices lead to lower demand.

3.3 Loss Function

In our model, we use Huber loss Eq. (2) [19], which is robust to outliers and reduces their impact on model training.

$$L(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta|y - \hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \quad (2)$$

For the region-level model, our model outputs the baseline charging demand. We train the model for this part using the actual charging of the region as the true value. For the station-level model, the left tower outputs the baseline charging demand, while the right tower predicts price elasticity, which is then used to calculate the final charging demand. In both towers, the real charging demand at the station is used as the ground truth to calculate the model loss.

4 Experiments

4.1 Dataset and Experimental Settings

Dataset. We used real-world data from 885 charging stations provided by Zeekr, an EVCS operator. The dataset includes hourly charging volumes and prices, totaling 21,240 records from 136 cities, along with basic station information such as latitude, longitude, and charging type. To facilitate dynamic pricing analysis, we aggregated the hourly data into time slots based on Zeekr’s time-of-use pricing strategy, resulting in 5,040 aggregated records. We collected 90 days of data from 2024.1.22 to 2024.4.2, including holidays like the Chinese New Year, to ensure representativeness and test our model’s performance. For the 90-day data, the first 70 days were used as the training set, the middle 10 days as the validation set, and the last 10 days as the test set.

Baseline Models. To validate our proposed model, we compare it with two established models: PAG [10] and PEM [17]. The PAG model uses a physics-informed, attention-based graph learning approach to predict charging demand. It employs a Graph Attention Network (GAT) for learning station adjacency matrices and a LSTM network for extracting temporal charging and price information. PAG also incorporates a priori elasticity knowledge and utilizes meta-learning for pre-training. The PEM model learns the dynamic price elasticity coefficient from price and demand sequences to predict precise demand. It features a Competitiveness Representation Module and a Multi-sequence Fusion Module to capture factors affecting the Price Elasticity of Demand (PED). Additionally, it uses a gating mechanism to learn PED from varying sparsity data.

Implementation Details. In our model, the Time Series Model segments data into overlapping windows of 28-day periods to capture monthly variations in

charging demand. Besides, normalization is used to scale the features to a consistent range. During meta-learning, station-level data is aggregated at the city level for pre-training the left tower model. We use the softplus [20] activation function to ensure non-negative outputs for predicting the price elasticity coefficient. We implemented our model using PyTorch [15], employing the Adam [18] optimizer with a learning rate of 0.001. The batch size was 512, with 300 epochs for region-level training and 1000 epochs for station-level training.

For the PAG model, we trained separate models for each city using station adjacency matrices. The meta-learning phase ran for 200 epochs, followed by 1500 epochs for station-level training with a batch size of 16. In the PEM model, we combined charging volume and price sequences as input for the time series module. We encoded city information, commercial area, station type, weather conditions, station price information, and holiday indicators as temporal factors. The training involved 2000 epochs, a batch size of 512, and a learning rate of 0.001. All models were trained using an NVIDIA Quadro RTX 5000 GPU.

4.2 Model Evaluation and Comparison

Evaluation Metrics. To evaluate our model’s performance, we employed the Weighted Absolute Percentage Error (WAPE) [21] to measure the performance:

$$WAPE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|} \quad (3)$$

where y_i represents the actual charging demand and \hat{y}_i represents the predicted charging demand.

Model Comparation. We compared our model with PEM and PAG. Table 1 shows the performance of our model outstanding compared to PEM and PAG. The overall average WAPE for all data shows that our model has the lowest error rate at 28.68%, compared to PEM’s 34.56% and PAG’s 29.76%.

Table 1. Model Comparison Results on Cities.

Model	WAPE (%)				
	Shenzhen	Beijing	Shanghai	Fuzhou	All Data Average
PEM	21.89	43.44	27.97	36.24	34.56
PAG	18.21	29.52	29.17	32.23	29.76
Ours	19.28	29.38	28.16	32.04	28.68

We analyzed the charging demand in several representative cities, as shown in Fig. 1, and categorized charging stations based on their price adjustment history and efficiency shown in Table 2. Stations with price adjustments indicate price

elasticity, we calculated the proportion of such stations in each city. Stations with fewer charging records are harder to predict. We classified stations with less than 45 kWh of charging volume as inefficient and others as efficient to calculate the proportion of inefficient stations in each city. In cities like Fuzhou, where there is substantial pricing adjustment data, our model effectively leverages the price elasticity coefficient, leading to better prediction accuracy. In contrast, in Shenzhen, with fewer pricing adjustments and no inefficient stations, our model's performance is close to but not superior to the PAG model.

Table 2. Cities Characteristics.

Characteristics	Shenzhen	Beijing	Shanghai	Fuzhou
Stations price adjust rate (%)	32.37	54.32	43.48	87.30
Inefficient Stations rate (%)	0.00	4.32	3.20	0.00

Overall, our model excels in predicting demand for stations with price changes and inefficient stations as shown in Table 3. The superior performance in these categories highlights our model's ability to handle stations with sparse charging data, and to adapt to varying pricing conditions across cities, improving demand prediction accuracy and effectively capturing the price adjustments.

Table 3. Model Comparison Results on Station Characteristics.

Model	WAPE (%)				
	Station Efficiency		Station Price		Average
	Inefficiency	Efficient	Price Fixed	Price Adjusted	
PEM	82.75	33.90	30.99	42.50	34.56
PAG	82.35	29.43	28.87	35.57	29.76
Ours	57.98	28.47	28.25	34.16	28.68

4.3 Ablation Experiments

We conducted an ablation study to assess the contribution of each key module in our model. Specifically, we evaluated two model variants:

Without Meta-learning Part: This variant excludes the region-level meta-learning component, which is designed to capture generalized regulation from city-level data to enhance the adaptability and robustness of the model for station-level predictions.

Without Right Tower: The right tower in the model aims to capture the station elasticity, modeling how the charging volume at each station is affected

Table 4. Table Ablation Experiment tables.

Model	WAPE
Without Meta-Learning Part	31.42
Without Right Tower	28.95
Ours	28.68

by price adjustments. This part omits the output from the left tower and only uses the baseline charging prediction as the model output.

As shown in Table 4, the ablation study indicates that both the right tower and meta-path components significantly contribute to the model's performance. The superior performance of our meta-learning model indicates its capability to handle sparse charging data. The prediction improvement from the right tower demonstrates its effectiveness in capturing the dynamics of charging demand influenced by price changes.

5 Conclusion

In this paper, we propose a novel approach that integrates baseline charging quantity and price elasticity to accurately predict the charging demand in EVCS. By considering both price-related and price-unrelated aspects, our model enables more precise demand charging and accounts for the impact of price changes. The meta-learning component further enhances our model's predictive accuracy. Besides, Extensive experiments on real-world datasets show that our model outperforms others in predicting charging demand.

In future work, we aim to optimize the right tower of our model to support dynamic pricing decisions by providing robust price elasticity. We will utilize causal inference methods to address the issue of sparse historical price adjustments and enhance prediction granularity for different types of charging stations, such as fast charging stations. Our goal is to maximize overall revenue while ensuring the efficient allocation of charging resources, contributing to the sustainability of the EV ecosystem.

References

1. Global EV Outlook 2024. <https://www.iea.org/reports/global-ev-outlook-2024>. Accessed 25 Oct 2023
2. Den Boer, Arnoud V.: Dynamic pricing and learning: historical origins, current research, and new directions. *Surv. Oper. Res. Manag. Sci.* **20**(1), 1–18 (2015)
3. Anderson, P.L., et al.: Price elasticity of demand. McKinac Center for Public Policy, 13 February 1997. Accessed October
4. Yi, T., et al.: Research on the spatial-temporal distribution of electric vehicle charging load demand: a case study in China. *J. Clean. Prod.* **242**, 118457 (2020)

5. Wang, S., et al.: Short-term electric vehicle charging demand prediction: a deep learning approach. *Appl. Energy* **340**, 121032 (2023)
6. Zheng, Z., et al.: Multi-energy load forecasting model based on bi-directional gated recurrent unit multi-task neural network. In: E3S web of Conferences, vol. 256. EDP Sciences, 2021
7. Su, S., et al.: Operating status prediction model at EV charging stations with fusing spatiotemporal graph convolutional network. *IEEE Trans. Transp. Electrif.* **9**(1), 114–129 (2022)
8. Hüttel, F.B., et al.: Deep spatio-temporal forecasting of electrical vehicle charging demand. arXiv preprint [arXiv:2106.10940](https://arxiv.org/abs/2106.10940) (2021)
9. Wang, S., et al.: Predicting electric vehicle charging demand using a heterogeneous spatio-temporal graph convolutional network. *Transp. Res. Part C: Emerg. Technol.* **153**, 104205 (2023)
10. Qu, H., et al.: A physics-informed and attention-based graph learning approach for regional electric vehicle charging demand prediction. arXiv preprint [arXiv:2309.05259](https://arxiv.org/abs/2309.05259) (2023)
11. Li, D., Lasenby, J.: Spatiotemporal attention-based graph convolution network for segment-level traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **23**(7), 8337–8345 (2021)
12. Yi, Y., Chen, Z., Li, R.: LSTM neural networks with attention mechanisms for accelerated prediction of charge density at onset condition of DC corona discharge. *IEEE Access* **10**, 124697–124704 (2022)
13. Bao, Z., et al.: Data-driven approach for analyzing spatiotemporal price elasticities of EV public charging demands based on conditional random fields. *IEEE Trans. Smart Grid* **12**(5), 4363–4376 (2021)
14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. PMLR (2017)
15. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019)
16. Dong, Y., Chawla, N.V., Swami, A.: Metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (2017)
17. Zhu, F., et al.: Modeling price elasticity for occupancy prediction in hotel dynamic pricing. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management (2022)
18. Kingma, D.P., Jimmy, B.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
19. Huber, P.J.: Robust estimation of a location parameter. In: Kotz, S., Johnson, N.L. (eds.) *Breakthroughs in Statistics*. Springer Series in Statistics, pp. 492–518. Springer, New York, NY (1992). https://doi.org/10.1007/978-1-4612-4380-9_35
20. Dugas, C., et al.: Incorporating second-order functional knowledge for better option pricing. *Adv. Neural Inf. Process. Syst.* **13** (2000)
21. De Myttenaere, A., et al.: Mean absolute percentage error for regression models. *Neurocomputing* **192**, 38–48 (2016)



BP-GNN-SBR: Behavior Progressive Graph Neural Networks for Session-Based Recommendation

Zekun Xu¹, Wenlong Wu¹, Zhanzuo Yin¹, Xinzhe Zhao¹, Junnan Zhuo¹,
and Bohan Li^{1,2,3(✉)}

¹ College of Artificial Intelligence and Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
bhli@nuaa.edu.cn

² Key Laboratory of Brain-Machine Intelligence Technology, Ministry of Education, Nanjing 211106, China

³ Key Laboratory of Intelligent Decision and Digital Operations, Ministry of Industrial and Information Technology, Nanjing, China

Abstract. Session-based recommendation leverages anonymous user interaction sequences to predict their next interaction. Most of the previous methods exploit a single user behavior for prediction, and some recent methods utilize multiple behaviors of users. However, they only consider the impact of behaviors on user preferences from a single perspective and do not explore the relationships between behaviors, resulting in inaccurate modeling. To tackle the issues, we propose **Behavior Progressive Graph Neural Networks (BP-GNN)** for Session-Based Recommendation, in which we model behavioral relationships from both macro and micro perspectives. In the macro view, we use item representations learned from clicking behavior as inputs to reinforce the learning process of buying behavior. In the micro view, input sequences are transformed into behavior-agnostic and operation sequences, with item representations learned for each. Then BP-GNN effectively captures information within the progressive relationships of behaviors. Experiments on three public datasets demonstrate that BP-GNN outperforms state-of-the-art models.

Keywords: Session-based recommendation · Graph neural networks · Multiple behaviors

1 Introduction

As one of the research directions of recommender systems, Session-based recommendation (SBR) [3,11] prioritizes short-term user interaction sequences and takes into account the user's current interests to provide more immediate recommendations. Numerous approaches have been proposed to address the challenge of SBR, including RNN-based methods [3,4], attention based methods

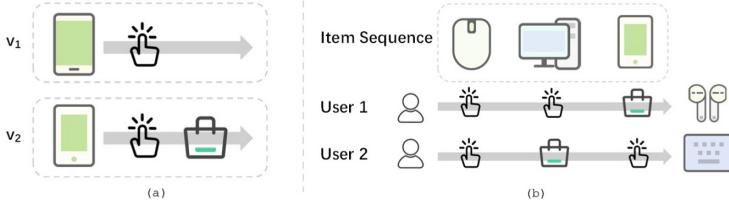


Fig. 1. A toy example of the progressive relationships of behaviors.

[7,9] and GNN-based methods [17,19]. However, most of them primarily center on single user behavior, overlooking the impact of multiple actions within a session.

In many platforms, the direct benefits are often related to single user behavior, such as the purchase behavior in e-commerce platforms, but the number of such interactions is small. Relying solely on such single behavior for recommendations may lead to a severe data sparsity issue. However, the click behavior in e-commerce platforms, which provides users with additional product information and aids them in making decisions, can effectively alleviate the data sparsity challenge. By incorporating multiple types of user behaviors [8,10,15], recommendation systems can learn a more holistic and informative user preference, ultimately enhancing the quality and relevance of the recommendations.

However, these methods only consider multiple behaviors at the macro level and deal with the different behaviors separately and then fuse them together, ignoring the progressive relationship of the behaviors. For example, users often click on a product before making a purchase. As shown in Fig. 1 (a), a user clicks on item v_1 but clicks and subsequently makes a purchase of item v_2 , it indicates that the user has a stronger preference for v_2 over v_1 . This reveals the fact that purchase behavior contains more user preference information than click behavior. Furthermore, the progressive relationship of behaviors is also important from a micro perspective. As shown in Fig. 1 (b), for the same item sequence, two users have different behavior sequences. User 1 clicks on some electronics and then buys a mobile phone, then his interest might be earphones related to the mobile phone, while user 2 buys a computer, then his preference might be a keyboard which is related to the computer. Therefore, it is also important to incorporate behavior transition patterns for user sequences in the micro perspective.

Taking these observations into account, we propose **Behavior Progressive Graph Neural Networks** for SBR, BP-GNN. Specifically, we consider the progressive relationship of behaviors at two levels, macro and micro. In the macro view, GNNs are used to learn the item representations of the click and purchase sequences, and we augment the item representation learning process of the purchase sequence with the item representation of the click sequence as input. In the micro view, the input sequences are transformed into behavior-agnostic sequences and operation sequences. We use GNNs to learn the item

representations and GRUs to learn the operation representations and then concatenate them together. In this way, we can capture the progressive relationships of behaviors at the micro level. Finally, we use the soft attention mechanism to learn the session representation for predicting the user’s next interaction item.

The main contributions of our work can be summarised as follows:

- We propose using the features learned from clicking behavior to enhance the learning process of purchasing behavior, in order to capture progressive behavioral information from the macro level.
- We additionally consider the progressive relationship of different behaviors in the micro perspective and propose a new model named behavior progressive graph neural network (BP-GNN) to fuse these two perspectives.
- We conduct sufficient experiments on three publicly available datasets to demonstrate that our model outperforms the state-of-the-art baseline model.

2 Related Work

RNNs are widely used in SBR [14] due to their powerful ability to capture sequence information. GRU4Rec [3] is the first work to use RNNs to learn item transition patterns. Li et al. [7] introduce attention mechanism into RNNs and use a hybrid encoder with attention mechanism to enhance the model’s performance. ISLF [13] captures users’ long-term and short-term interests and use RNNs to learn their interest changes. Many methods [2,6] use GNNs to capture more complex relationships in sequences. SR-GNN [17] is the first to use GNNs to aggregate item information within a session. Zhang et al. [21] build a heterogeneous hypergraph and then use co-guided learning to establish relationships between prices and user preferences. DAT-MDI [1] uses a global graph to capture contextual information in different sessions between multiple domains.

Many multi-behavior models have been proposed to exploit behavior information [18,20]. MBGCN [5] captures various intensities of different behaviors and learns semantic information about multiple behaviors. Xuan et al. [20] propose using knowledge graphs to assist in capturing multiple behavioral dependencies and obtaining personalized user preferences. MGNN-Spred [15] build a multi relation item graph to capture item relations in the global level. Liang et al. [8] use GNN to learn information from different behaviors separately and propose a sparse self-attention mechanism to mitigate noise in the sequence.

3 Preliminaries

Let $V = \{v_1, v_2, \dots, v_n\}$ be the set of all items. An anonymous session s is represented as $s = [v_{s,1}, v_{s,2}, \dots, v_{s,l}]$, where $v_{s,i}$ denotes the i -th item interacted by user. Then s is converted into four types of sequences, i.e. $s(c)$, $s(b)$, $s(a)$ and $s(o)$. $s(c) = [v_{c,1}, v_{c,2}, \dots, v_{c,|s(c)|}]$ denotes the click sequence in session s , and $s(b)$ denotes the buy sequence. $s(a) = [v_1, v_2, \dots, v_l]$ denotes the behavior-agnostic sequence. $s(o) = [o_1, o_2, \dots, o_l]$ denotes the operation sequence, where o_i denotes the behavior type of item v_i (i.e. click or buy).

For a sequence s , we construct a session graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ where $\mathcal{V}_s \subseteq V$ denotes the set of items in sequence s , and the edge $(v_i, v_{i+1}) \in \mathcal{E}_s$ represents that there is an edge from node v_i to v_{i+1} . In addition, a self-loop is added to each node [16]. After that, we set up four types of edges for the graph \mathcal{G}_s , i.e. e_{in} , e_{out} , e_{in-out} and e_{self} . For nodes v_i and v_j , e_{in} means that there is only one edge from v_i to v_j , e_{out} denotes that there is only one edge from v_j to v_i , e_{in-out} implies that there is a bidirectional edge between v_i and v_j , and e_{self} means that node v_i has a self-loop edge. According to the above construction, for a session s , we can construct three graphs: click sequence graph \mathcal{G}_c , buy sequence graph \mathcal{G}_b and behavior-agnostic sequence graph \mathcal{G}_a .

Figure 2 shows the overall structure of our BP-GNN model.

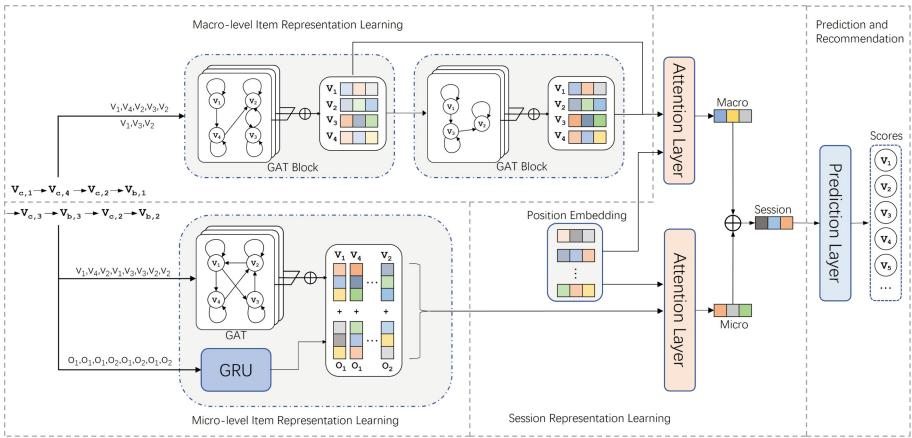


Fig. 2. The overall framework of our proposed method.

4 The Proposed Method

4.1 Macro-level Item Representation Learning

From a macro perspective, users always interact with items in a certain order i.e. clicking and then purchasing. In such a progressive pattern, the latter behavior often exhibits more important user preferences than the previous behavior. Therefore, the item representation learned from the previous behavior can be used to optimize the embedding learning of the latter behavior.

We utilize GAT to aggregate item information on previously constructed graphs, since different nodes have different importance to the current node in the session graph. Firstly, the attention coefficients can be calculated as:

$$p_{ij} = \text{LeakyReLU}(\mathbf{r}_{e_{ij}}^\top (\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_j^{(l-1)})), \quad (1)$$

where p_{ij} means the importance of node v_j to node v_i , $\mathbf{x}_i^{(l-1)}$ is the representation of item v_i in the $(l-1)$ -th layer, $\mathbf{r}_{e_{ij}}$ denotes four trainable vectors representing four different edges, \odot is the element-wise multiplication, \mathcal{N}_{v_i} denotes the set of neighbours of node v_i . Afterwards, we use the softmax function to compute the normalised attention weights α_{ij} to make the attention coefficients comparable, and calculate the representation of v_i of the l -th layer:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{r}_{e_{ij}}^\top (\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_j^{(l-1)})))}{\sum_{v_k \in \mathcal{N}_{v_i}} \exp(\text{LeakyReLU}(\mathbf{r}_{e_{ik}}^\top (\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_k^{(l-1)})))}. \quad (2)$$

$$\mathbf{x}_i^l = \sum_{v_k \in \mathcal{N}_{v_i}} \alpha_{ij} \mathbf{x}_j^{(l-1)}. \quad (3)$$

We consider the above method as a GAT block and through a GAT block, we can obtain the click item representation \mathbf{x}_i^c , i.e. \mathbf{x}_i^l , whose input is the learnable initialization embedding i.e. \mathbf{x}_i^0 . And then the click sequence with the node representation can be denoted as $\mathbf{v}_c = [\mathbf{x}_1^c, \mathbf{x}_2^c, \dots, \mathbf{x}_{|s(c)|}^c]$. Using the same method, we can also obtain the purchase item representation \mathbf{x}_i^b , but the difference is that the input of this GAT block is the click item representation \mathbf{x}_i^c obtained in the previous step. After that, the buy sequence can be denoted as $\mathbf{v}_b = [\mathbf{x}_1^b, \mathbf{x}_2^b, \dots, \mathbf{x}_{|s(b)|}^b]$.

Through this method, BP-GNN can learn macro progressive relationships by using the learned embedding of the previous behavior as input to the latter.

4.2 Micro-level Item Representation Learning

Next, we learn the item representation from a micro level. From a micro perspective, the sequence of interactions between users and items tends to exhibit different transition patterns due to the influence of multiple behaviors. In order to effectively learn the impact of multi behaviors on item transitions, we transform the input sequences into behavior-agnostic item sequences $s(a)$ and operation sequences $s(o)$. Then, we use GAT to learn item embeddings and GRU to learn operational embeddings. For an item sequence, the transition pattern is complex, and not only does the previous item affect the current item, but the later item also has an effect. Therefore, we use GAT to learn this complex transition pattern. For an operation sequence, what we need is the progressive information about the behaviors, and GRU can capture this information well.

For a behavior-agnostic sequence, a GAT block is used to learn its item representation, whose input is the learnable initialization embedding. And then the behavior-agnostic sequence with the item representation can be represented as $\mathbf{v}_a = [\mathbf{x}_1^a, \mathbf{x}_2^a, \dots, \mathbf{x}_l^a]$ where l is the sequence length. We then use GRU to learn the embedding of the operation sequences. The input of the GRU block is an operation sequence $s(o)$. For each operation o_i within $s(o)$, the initial embedding

\mathbf{o}_i^0 is retrieved from the operation embedding matrix. Subsequently, the GRU is employed to obtain the updated embedding:

$$\mathbf{z}_i = \sigma(\mathbf{W}^z \mathbf{o}_i + \mathbf{U}^z \mathbf{h}_{i-1}), \quad (4)$$

$$\mathbf{r}_i = \sigma(\mathbf{W}^r \mathbf{o}_i + \mathbf{U}^r \mathbf{h}_{i-1}), \quad (5)$$

$$\tilde{\mathbf{h}}_i = \tanh(\mathbf{W}^h \mathbf{o}_i + \mathbf{W}^h (\mathbf{h}_{i-1} \odot \mathbf{r}_i)), \quad (6)$$

$$\mathbf{o}_i = \mathbf{h}_i = (1 - \mathbf{z}_i) \odot \tilde{\mathbf{h}}_i + \mathbf{z}_i \odot \mathbf{h}_{i-1}, \quad (7)$$

where \mathbf{z}_i is the update gate, \mathbf{r}_i is the reset gate, \mathbf{W} and \mathbf{U} are trainable parameters and σ denotes the sigmoid activation function, \mathbf{h}_i denotes the hidden state in the i -th step. After that, the operation sequence is obtained using the learned embeddings of all operations, represented as $\mathbf{v}_o = [\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_l]$. Then, we concatenate \mathbf{v}_a and \mathbf{v}_o to obtain the final micro-level item representation:

$$\mathbf{v}_{ao} = [\mathbf{x}_1^{ao}, \mathbf{x}_2^{ao}, \dots, \mathbf{x}_l^{ao}] = [\mathbf{x}_1^a \parallel \mathbf{o}_1, \mathbf{x}_2^a \parallel \mathbf{o}_2, \dots, \mathbf{x}_l^a \parallel \mathbf{o}_l], \quad (8)$$

where \parallel is the concatenation operation. By employing the aforementioned methods, we can ensure that sequences with the same items may have different representations when their operation sequences are different. This method enables us to identify and learn user preferences at a micro level.

4.3 Session Representation Learning

Session representations can be obtained by aggregating information from both macro and micro level item representations. For an updated sequence $\mathbf{v} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$, we now present how to obtain the session representation.

Note that items in different locations have different impacts on user preferences, in other words, the later the interaction items, the greater the impact on user preferences. Hence, we attach the reversed position embedding to each session item as follows:

$$\mathbf{t}_i = \sigma(\mathbf{W}_1[\mathbf{x}_i \parallel \mathbf{p}_i] + \mathbf{b}_1), \quad (9)$$

where \mathbf{p}_i is the reversed position embedding of the i -th position, $\mathbf{W}_1 \in \mathcal{R}^{d \times 2d}$ and $\mathbf{b}_1 \in \mathcal{R}^d$ are trainable parameters of nonlinear transformation and σ is an activation function, here we use \tanh . Next, we use soft attention mechanism to get the session representation. Considering that the last item in the session reflects strong current preference information of the user, it is utilized to guide the computation of the attention value. Specifically, given an item representation \mathbf{t}_i , its attention weight is computed as

$$\beta_i = \mathbf{a}^\top \sigma(\mathbf{W}_2 \mathbf{t}_m + \mathbf{W}_3 \mathbf{t}_i + \mathbf{b}_2), \quad (10)$$

where $\mathbf{W}_2, \mathbf{W}_3 \in \mathcal{R}^{d \times d}$ and $\mathbf{a}, \mathbf{b}_2 \in \mathcal{R}^d$ are trainable parameters and σ is the sigmoid function. \mathbf{t}_m represents the last item of the current session. At last, the sequence representation can be get as follows:

$$\mathbf{s}_v = \sum_{i=1}^m \beta_i \mathbf{t}_i. \quad (11)$$

Table 1. Basic statistics of the datasets.

Dataset	Yoochoose	Tmall	Cosmetics
items	24,239	192,319	41,193
training data	163,005	215,690	325,284
validation data	12,985	12,607	16,906
test data	25,971	25,214	33,812

For macro-level item representation sequences \mathbf{v}_c and \mathbf{v}_b , the above method is utilized to obtain sequence representations \mathbf{s}_c and \mathbf{s}_b . Then we add them together to get the macro-level sequence representation \mathbf{s}_{macro} . In the same way, for micro-level item representation sequences \mathbf{v}_{ao} , the sequence representations can be obtained as \mathbf{s}_{micro} . Finally, an aggregation function is used to fuse macro and micro level sequence representations to obtain the final session representation:

$$\mathbf{s} = \text{Agg}(\mathbf{s}_{macro}, \mathbf{s}_{micro}) = \mathbf{s}_{macro} + \mathbf{s}_{micro}, \quad (12)$$

where $\text{Agg}(\cdot)$ denotes the aggregation function. In contrast to prior work [8, 10] employing gating mechanism as aggregation function, we adopt sum pooling, a simpler but more efficient approach. In Subsect. 5.4, we delve into the varying impacts of different aggregation functions on our model.

4.4 Prediction and Recommendation

We then compute the recommendation probability $\hat{\mathbf{y}}_i$ for a candidate item v_i by applying the inner product followed by a softmax function:

$$\hat{\mathbf{y}}_i = \text{softmax}(\tilde{\mathbf{y}}_i) = \frac{\exp(\mathbf{s}^\top \mathbf{x}_i)}{\sum_{i=1}^n \exp(\mathbf{s}^\top \mathbf{x}_i)}. \quad (13)$$

The cross-entropy loss is used to guide the model learning and \mathbf{y}_i denotes the one-hot encoding vector of the ground truth item:

$$\mathcal{L}(\hat{\mathbf{y}}) = - \sum_{i=1}^n \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i), \quad (14)$$

5 Experiments

5.1 Datasets and Preprocessing

We apply three real-world datasets to evaluate our model, known as Yoochoose¹, Tmall² and Cosmetics³. Following [10, 15], we preprocess the three

¹ <http://2015.recsyschallenge.com/challenge.html>.

² <https://tianchi.aliyun.com/dataset/42>.

³ <https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-cosmetics-shop>.

Table 2. Performance(%) of all comparison methods on three datasets.

Methods	Yoochoose			Tmall			Cosmetics		
	H@20	M@20	N@20	H@20	M@20	N@20	H@20	M@20	N@20
POP	0.708	0.139	0.266	0.917	0.462	0.560	2.337	0.219	0.429
Item-KNN [12]	8.748	2.374	3.767	1.965	0.339	0.682	7.143	1.684	2.916
GRU4Rec [3]	12.616	2.925	5.019	2.429	0.698	1.071	11.956	2.852	5.102
STAMP [9]	12.662	3.287	5.327	2.656	0.897	1.289	12.238	3.004	5.335
SR-GNN [17]	12.959	3.588	5.634	2.998	1.045	1.482	13.148	3.379	5.523
GC-SAN [19]	12.756	3.228	5.386	3.032	0.968	1.455	12.172	3.102	5.473
MGNN-SPred [15]	14.417	3.406	5.790	5.701	1.789	2.655	14.181	3.674	5.945
BA-GNN [8]	14.835	3.940	6.382	6.017	2.083	2.949	14.835	3.940	6.382
BGNN [10]	19.167	5.267	8.332	8.691	2.918	4.205	18.987	5.673	8.819
BP-GNN	18.654	6.824	9.404	9.264	3.401	4.706	19.469	7.051	9.801

datasets as follows. For a given session, it contains a click sequence $s(c) = [v_{c,1}, v_{c,2}, \dots, v_{c,|s(c)|}]$ and a buy sequence $s(b) = [v_{b,1}, v_{b,2}, \dots, v_{b,|s(b)|}]$. We split the behavior sequences to generate the sequences and labels. For an item $v_{b,i}$ in buy sequence $s(b)$, we regard $[v_{b,1}, v_{b,2}, \dots, v_{b,i-1}]$ as the input sequence and $v_{b,i}$ as the target. For the click sequence $s(c)$, we only keep the items before the target item to avoid the click sequence seeing the label. Afterwards, we also set the maximum sequence length to L. Table 1 shows the basic information of the three datasets.

5.2 Parameter Setup

Following the previous work [10, 16], the dimension of embedding vectors is set to 64 for a fair comparison. We set the maximum epoch to 20 and use an early stopping strategy to prevent overfitting. We set the maximum sequence length to 4 for Yoochoose and 3 for Tmall and Cosmetics, and the number of layers of GAT to 1. We utilize Adam optimizer with the initial learning rate of 0.001 and decay the learning rate after every three epochs. The L2 penalty is set to 10^{-5} .

5.3 Overall Comparison

We adopt three ranking metrics by following previous work [10]: HR@20(H@20), MRR@20 (M@20), NDCG@20(N@20) to evaluate the recommendation performance of all models in Table 2. Here, we can make the following observations:

GNN-based approaches outperform the RNN-based models, suggesting that GNNs can capture complex transition relationships within a session. This also reflects that RNN-based methods are difficult to leverage information outside of the transition pattern. The performance of multi behavior methods is superior to other methods. BA-GNN outperforms MGNN-SPred because it introduces a

sparse self-attention mechanism to mitigate the noise. BGNN achieves better performance, demonstrating the effectiveness of integrating homogeneous and heterogeneous behavioral transition information.

Our proposed model achieves optimal results on all datasets. The main reason is that we consider the progressive relationships of behaviors and learn the user’s multi behavior preferences from two perspectives: macro and micro. However, The HR result of our model is slightly lower than those of BGNN on Yoochoose dataset, which may be due to the fact that BGNN utilises all sessions to build the graph, which better takes into account global user preferences whereas we pay more attention to considering the current preferences of users.

Table 3. Ablation study on three datasets.

Methods	Yoochoose			Tmall			Cosmetics		
	H@20	M@20	N@20	H@20	M@20	N@20	H@20	M@20	N@20
w/o micro	17.692	5.169	7.879	7.692	2.774	3.357	17.769	5.808	8.389
w/o progress	17.853	6.268	9.023	8.883	2.944	4.227	18.241	6.599	9.252
w/ gating	17.500	6.592	9.048	8.122	3.026	4.164	18.584	6.640	9.207
w/ concat	16.731	6.736	9.132	7.106	3.426	4.257	17.994	6.557	9.070
BP-GNN	18.654	6.824	9.404	9.264	3.401	4.706	19.469	7.051	9.801

5.4 Ablation Study

Impact of Different Modules. We defined *w/o micro* to represent BP-GNN without micro behaviors, and *w/o progress* to represent the model without progressive relationship of behaviors at the macro-level. Table 3 exhibits that *BP-GNN w/o micro* achieves the worst results overall. This suggests that micro-level behavioral transfer patterns of user sequences provide well information on user preferences. *BP-GNN* performs better than *BP-GNN w/o progress*, proving that there is a progressive relationship between behaviors, i.e., the latter behavior reveals user preferences more accurately than the former.

Impact of Feature Aggregation. As we use macro feature encoder and micro feature encoder, it is meaningful to compare BP-GNN with different aggregation operations i.e., gating and concatenation. From Table 3 we can see that sum pooling used in BP-GNN has the best results. On the contrary, the gating mechanism used in the previous works [8,15] does not work very well. This also shows that more parameters may not bring better results.

5.5 Impact of Parameters

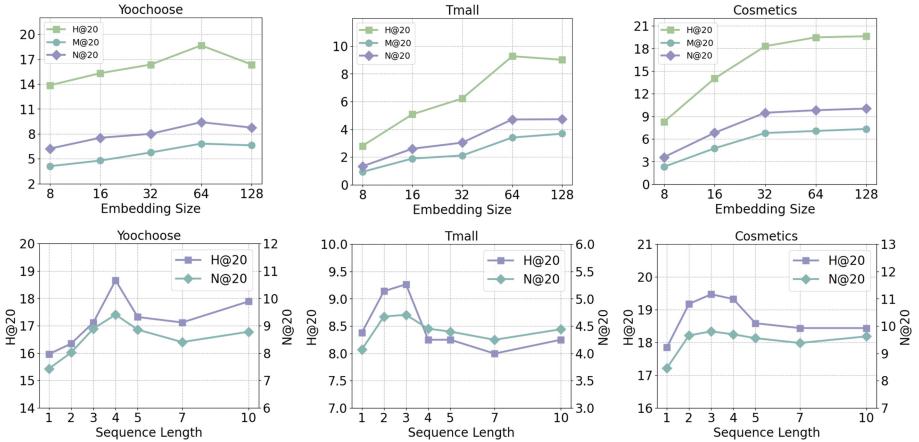


Fig. 3. Impact of different parameters on three datasets.

Impact of Embedding Size. We adjust the embedding dimension of the model. Figure 3 shows that when the embedding size increases, the model performance improves accordingly. However, when the embedding size is 128, the results of the Yoochoose dataset drop. This is because large embedding dimension will lead to overfitting.

Impact of Sequence Length. As the length of the sequence increases, it can be observed from Fig. 3 that the indicators initially show an upward trend, then begin to decline, and finally turn to be stable. From this we can see that when the sequence length is insufficient, the model fails to capture enough user preference information from the sequence. On the contrary, the model has difficulty in capturing the short-term preference of the user, or there is too much noise information in the sequence that leads to poor results.

6 Conclusion

In this paper, we propose a novel model named BP-GNN for session-based recommendation. Our model effectively exploits the progressive relationship between behaviors from both macro and micro perspectives. At the macro level, we use the learned item representations of the previous behavior to enhance the learning process of the latter; at the micro level, we transform the input sequences into behavior-agnostic sequences and operation sequences to learn behavior transition patterns. Finally, the soft attention mechanism is used to fuse the information from both levels to make recommendations for users. Moreover, sufficient experiments are conducted on three real-world datasets to demonstrate the effectiveness of our model.

Acknowledgements. This work is supported in part by the “14th Five-Year Plan” Civil Aerospace Pre-Research Project of China under Grant No. D020101, the Natural Science Foundation of China No. 62302213, Innovation Funding of Key Laboratory of Intelligent Decision and Digital Operations No. NJ2023027, Ministry of Industrial and Information Technology Project of Hebei Key Laboratory of Software Engineering, No. 22567637H, the Natural Science Foundation of Jiangsu Province under Grant No. BK20210280.

References

1. Chen, C., Guo, J., Song, B.: Dual attention transfer in session-based recommendation with multi-dimensional integration. In: SIGIR, pp. 869–878 (2021)
2. Han, Q., Zhang, C., Chen, R., Lai, R., Song, H., Li, L.: Multi-faceted global item relation learning for session-based recommendation. In: SIGIR, pp. 1705–1715 (2022)
3. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint [arXiv:1511.06939](https://arxiv.org/abs/1511.06939) (2015)
4. Jannach, D., Ludewig, M.: When recurrent neural networks meet the neighborhood for session-based recommendation. In: RecSys, pp. 306–310 (2017)
5. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: SIGIR, pp. 659–668 (2020)
6. Lai, S., Meng, E., Zhang, F., Li, C., Wang, B., Sun, A.: An attribute-driven mirror graph network for session-based recommendation. In: SIGIR, pp. 1674–1683 (2022)
7. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: CIKM, pp. 1419–1428 (2017)
8. Liang, Y., Song, Q., Zhao, Z., Zhou, H., Gong, M.: BA-GNN: behavior-aware graph neural network for session-based recommendation. FCS **17**(6), 176613 (2023)
9. Liu, Q., Zeng, Y., Mokhosi, R.: Stamp: short-term attention/memory priority model for session-based recommendation. In: SIGKDD, pp. 1831–1839 (2018)
10. Luo, J., He, M., Pan, W., Ming, Z.: BGNN: behavior-aware graph neural network for heterogeneous session-based recommendation. FCS **17**(5), 175336 (2023)
11. Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., De Rijke, M.: Repeatnet: a repeat aware neural recommendation machine for session-based recommendation. In: AAAI, vol. 33, pp. 4806–4813 (2019)
12. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295 (2001)
13. Song, J., Shen, H., Ou, Z., Zhang, J., Xiao, T., Liang, S.: ISLF: interest shift and latent factors combination model for session-based recommendation. In: IJCAI, pp. 5765–5771 (2019)
14. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: RecSys, pp. 17–22 (2016)
15. Wang, W., et al.: Beyond clicks: modeling multi-relational item graph for session-based target behavior prediction. In: WWW, pp. 3056–3062 (2020)
16. Wang, Z., Wei, W., Cong, G., Li, X.L., Mao, X.L., Qiu, M.: Global context enhanced graph neural networks for session-based recommendation. In: SIGIR, pp. 169–178 (2020)
17. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI, vol. 33, pp. 346–353 (2019)
18. Xia, L., Xu, Y., Huang, C., Dai, P., Bo, L.: Graph meta network for multi-behavior recommendation. In: SIGIR, pp. 757–766 (2021)

19. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI, vol. 19, pp. 3940–3946 (2019)
20. Xuan, H., Liu, Y., Li, B., Yin, H.: Knowledge enhancement for contrastive multi-behavior recommendation. In: WSDM, pp. 195–203 (2023)
21. Zhang, X., et al.: Price does matter! modeling price and interest preferences in session-based recommendation. In: SIGIR, pp. 1684–1693 (2022)



Exploring Simple Architecture of Just-in-Time Compilation in Databases

Haoran Ning¹, Bocheng Han¹, Zhengyi Yang¹ (✉) Kongzhang Hao¹,
Miao Ma¹, Chunling Wang², Boge Liu², Xiaoshuang Chen², Yu Hao², Yi Jin²,
Wanchuan Zhang², and Chengwei Zhang²

¹ The University of New South Wales, Sydney, Australia

{haoran.ning, bocheng.han, zhengyi.yang, k.hao, miao.ma}@unsw.edu.au

² Data Principles (Beijing) Technology Co., Ltd., Beijing, China

{chunling.wang, boge.liu, xiaoshuang.chen, yu.hao, yi.jin, wanchuan.zhang,
chengwei.zhang}@enmotech.com

Abstract. Just-in-Time (JIT) compilation is an effective technique for enhancing query execution in modern relational databases, and it has gained increasing attention from academia and industry in recent years. However, the architectures of state-of-the-art JIT-based database systems are often complex, leading to challenges and limitations when adopted for commercial use. In this paper, we present an industrial view to JIT compilation for relational databases, emphasizing practicality and applicability. Our focus is on minimizing engineering effort, simplifying testing, and ensuring seamless integration with existing database ecosystems. We achieve these goals by adhering to three core principles: a simple, lightweight architecture; reuse of existing technologies and frameworks, particularly LLVM; and strong extensibility and compatibility. We demonstrate the feasibility and potential of this approach through an initial exploration using LLVM’s mature JIT compilation capabilities to translate TPC-H database queries into optimized machine code. This proof-of-concept implementation shows the promise of our approach, pivoting the way for a comprehensive database system that leverages a lightweight yet powerful JIT compilation framework for real-world applications.

Keywords: JIT · RDBMS · LLVM · TPC-H

1 Introduction

Relational databases play a foundational role in data management across various industries, structuring data into tables and enabling complex queries through inter-table relationships. They power many applications in critical sectors such as banking, trading, manufacturing, and healthcare. However, with the continual evolution of hardware, including the expansion of memory capacity and the enhancement of CPU capabilities [4], there arises a pressing demand for enhanced query execution performance in real-time, data-intensive environments.

Traditionally, database queries have been executed using the iterator model, which, however, is associated with limitations such as interpretation overhead and low utilization of CPU cache. In recent years, Just-in-Time (JIT) compilation has garnered significant attention for its potential to improve the efficiency of query execution in relational databases. JIT compilation dynamically compiles SQL queries into optimized machine instructions at the time of execution, thereby bypassing the overhead associated with traditional interpretive execution and improving CPU cache utilization, leading to accelerated query execution.

In various fields, JIT compilers are already widely utilized in diverse applications, ranging from optimizing Python code execution with tools like PyPy, Numba, and Cython to enhancing Java bytecode in the Hotspot JVM [3, 16, 17]. Similarly, in the realm of databases, the dynamic nature of JIT compilation effectively addresses the limitations of traditional query processing methods. It facilitates on-the-fly optimization and compilation of query plans based on runtime conditions such as data size and distribution, available hardware resources, and the specific query operators used. By dynamically adapting to these factors, JIT compilation can yield significant performance enhancements, particularly for complex or frequently executed queries.

Many modern database systems have embraced JIT compilation to boost query execution performance. They fall into two categories [11]: Query Plan Execution (QPE) and Expression (EXP). QPE systems, such as HyPer [8], compile entire query plans into machine code, optimizing across the execution pipeline. In contrast, EXP systems, exemplified by PostgreSQL [12], compile individual expressions for specific optimizations. While both approaches offer benefits, QPE holds greater promise in enhancing performance by fully leveraging the power of JIT compilation. Notable JIT-based systems include PostgreSQL [12], HyPer [8], Umbra [14], ClickHouse [7], and Mutable [6], with HyPer being a representative pioneer.

HyPer employs a multi-layered architecture, where incoming SQL queries undergo initial parsing and optimization [9, 10, 13, 15]. Subsequently, HyPer's code generator translates these optimized queries into a customized intermediate representation (IR). This IR is then passed to a dynamic compiler to produce highly efficient machine code tailored for the specific query and the underlying hardware. The generated code is then executed directly, bypassing the traditional interpretation step and significantly accelerating query processing. Additionally, HyPer introduces an adaptive execution model that intelligently determines whether to execute a query using interpretation or compilation based on runtime factors such as query complexity and data characteristics.

Limitations. While HyPer demonstrates commendable performance, it also introduces certain limitations when applying it in industrial practice:

(1) *High Engineering Efforts:* Implementing HyPer's architecture requires substantial expertise across diverse domains, including developing a customized intermediate representation (IR), managing intricate adaptive execution logic, and coordinating fine-grained parallelism. This complexity demands in-depth

knowledge of compiler techniques, operating systems, and storage systems, necessitating a team of domain experts. However, the intricate design leads to a sizable codebase, complex inter-component interactions, prolonged development time, and elevated maintenance costs, potentially hindering adoption by commercial entities prioritizing time-to-market and cost-effectiveness.

(2) *Software Quality Assurance*: In HyPer, SQL queries undergo a multi-stage pipeline to convert them into low-level machine code, involving a long compilation chain. This poses significant challenges for comprehensive testing and verification. Rigorous testing is crucial for ensuring software reliability and adherence to quality standards in commercial products. However, the absence of standardized tools for tracing the execution path or examining intermediate results complicates systematic validation of the generated code. Additionally, the intricate nature of HyPer's internal mechanisms, including adaptive execution and morsel-driven parallelism, adds complexity to creating exhaustive test cases that cover all potential code paths and edge scenarios.

(3) *Limited Extendability*: HyPer's intricate architecture, characterized by tightly coupled components and specialized mechanisms, poses challenges for extending its functionality and integrating it with other systems. The complex interactions between its adaptive execution, dynamic compilation, and morsel-driven parallelism make it challenging to introduce new features or modify existing ones without extensive refactoring. This lack of modularity may hinder the adoption of HyPer in environments where flexibility and customization are crucial. Moreover, HyPer's reliance on complex components and interfaces can limit its compatibility with existing tools and frameworks, further impeding its integration into broader data processing pipelines.

Aim. Motivated by the aforementioned limitations, we aim to envision a system that explores the benefits of JIT compilation while sidestepping the complexity of existing academic systems like HyPer. Our key goals include:

(1) *Low Engineering Effort*: The system should prioritize a simple design that minimizes engineering effort and facilitates a clear understanding of the system's internal workings.

(2) *Ease of Testing*: The system should be designed with software quality in mind, enabling easy testing to ensure the correctness and reliability.

(3) *Ecosystem Compatible*: The system should be modular and extendable, enabling seamless integration with other database components and external tools in existing database Ecosystem.

Roadmap. To verify the feasibility of constructing a lightweight yet powerful JIT compilation framework, we adhere to three core principles:

(1) *Simple, Lightweight Architecture*: Our primary focus is on a straightforward design that minimizes engineering effort and promotes a clear comprehension of the system's internal mechanisms. Diverging from HyPer's inclusion of intricate

components like customized IR and adaptive execution mechanisms, we advocate for simplicity. This approach not only streamlines development and maintenance but also improves debuggability. By mitigating complexity, we establish a more resilient and dependable system, less susceptible to unforeseen failures.

(2) Re-use of Existing Technologies and Frameworks: We aim to leverage established technologies and frameworks to circumvent redundant development. This approach not only cuts down on development time and effort but also enables us to capitalize on the thorough testing and refinement. Additionally, leveraging existing components grants access to a plethora of monitoring, profiling, and debugging tools, thereby augmenting the system's reliability.

(3) Extendability and Compatibility: Our system is crafted with a steadfast commitment to modularity and adherence to established standards, facilitating integration with existing database components and other tools within the ecosystem. This design philosophy empowers us to capitalize on pre-existing functionalities such as buffer management and query optimization, thereby mitigating redundant development endeavors. Moreover, compatibility with industry standards streamlines integration with various systems and paves the way for future enhancements by seamlessly incorporating new components and features.

Contributions. In this initial exploration, we leverage the widely-used LLVM compiler infrastructure, specifically its mature JIT compilation capabilities. Our approach involves implementing database queries in TPC-H into C++ representations, which are then compiled into optimized machine code using LLVM's JIT compiler. Our choice of LLVM is strategic. As a mature and widely adopted framework, LLVM offers a robust foundation for our JIT compiler, minimizing engineering effort. Moreover, LLVM provides a rich set of tools for debugging, ensuring that our system remains transparent and easy to troubleshoot. The extensive use of LLVM in the industry further strengthens our confidence in its reliability and performance. This initial exploration serves as a proof of concept, demonstrating the feasibility and potential of our LLVM-based approach. We anticipate that this simple architecture will not only yield performance gains but also reduce engineering effort, improve software quality, and enhance extendibility. Subsequent work will build upon these foundations to develop a comprehensive database system that fully embodies the benefits of our lightweight JIT compilation framework.

Paper Organization. The rest of this paper is organized as follows. Section 2 introduces preliminaries. Section 3 presents our approach's architecture. Empirical evaluations are in Sect. 4, followed by future work in Sect. 5 and conclusion in Sect. 6.

2 Preliminary

Compilation Techniques: AOT and JIT. Ahead-of-Time (AOT) compilation translates code into native machine code before execution, as often used

for languages like C and C++ [18]. This approach can reduce runtime overhead, particularly for programs with short execution times. However, it lacks the ability to leverage runtime profile data for dynamic optimizations. Just-In-Time (JIT) compilation, conversely, translates code during execution, enabling optimizations based on real-time program behavior. This is crucial for database systems where query patterns and data distributions can vary significantly. JIT compilation, while potentially introducing runtime overhead, can lead to superior performance in such dynamic environments.

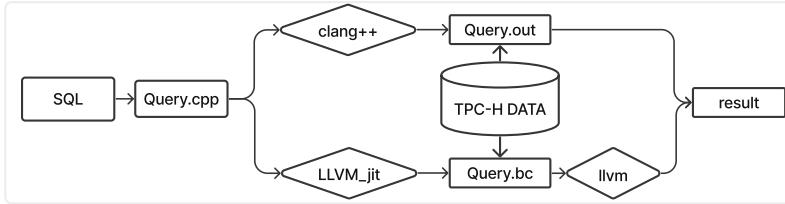


Fig. 1. From SQL To C++ Architecture

TPC-H Benchmark. The TPC-H benchmark is a standardized tool used to assess the performance of database systems, focusing on their ability to handle complex analytical queries. This benchmark tests a database's efficiency in managing real-world operations, which include multi-table joins and extensive data aggregations [2]. By applying the TPC-H benchmark, organizations can measure important performance metrics such as query execution time and system throughput. These metrics are crucial for understanding how well a database handles complex queries, ensuring that comparisons across different systems are consistent and fair. And we access handwritten C++ code and traditional database systems by the TPC-H benchmark.

LLVM. LLVM (Low Level Virtual Machine) is a compiler infrastructure project that supports dynamic compilation [5], enabling rapid code compilation and effective optimization. It is a key tool for achieving high performance in database systems due to its ability to generate hardware-specific code and integrate seamlessly with existing C++ codebases.

LLI. LLI is a command-line tool within LLVM specifically designed to execute programs written in LLVM Intermediate Representation (IR) using LLVM's JIT compilation capabilities [1]. This facilitates rapid development and testing of LLVM code, making it particularly useful for performance testing and iterative development in compiler-related projects.

3 Architecture

In order to assess the feasibility of applying JIT compilation to database systems and explore its benefits, we present the architecture of our experimental framework.

Overview. To test the application of JIT compilation in a database, our general idea is to first translate SQL statements into corresponding C++ code. Then, the C++ code is compiled using two different methods discussed as follows in Fig. 1:

Static Compilation. The C++ code is compiled into an executable file (Query.out) using the clang++ compiler. This executable reads TPCH data at runtime, executes the corresponding query, and outputs the results.

JIT Compilation. The C++ code is compiled into an intermediate representation file (Query.bc) using the LLVM JIT compiler. The LLVM JIT engine then runs this intermediate representation file, reads the TPCH data, executes the corresponding query, and outputs the results.

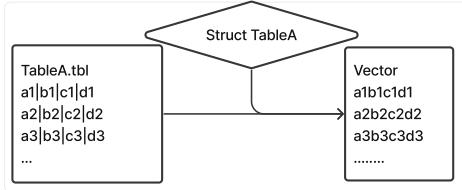
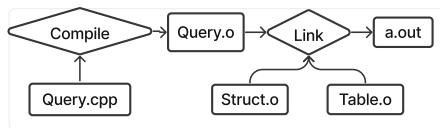
The main reason we chose these two compilation methods is their difference in compilation timing: static compilation compiles all code before the program runs, while JIT compilation compiles the code at runtime. By comparing the performance of these two methods, we can evaluate the application effect of JIT in database queries.

By adhering to this approach, we achieve the three primary goals outlined in our aim. First, the direct translation of SQL to C++ and the utilization of established compilers (clang++, LLVM JIT) significantly reduce the engineering complexity, making the system easily comprehensible and maintainable. Moreover, the modular structure, with well-defined stages for SQL translation, compilation, and query execution, allows for granular testing and validation. Standard debugging tools can be readily employed to trace execution paths and examine intermediate results, ensuring the reliability and correctness of the generated code. Lastly, the choice of C++ as the intermediate representation and the compatibility with multiple compilers promote extensibility and seamless integration with existing database ecosystems.

From SQL to C++. In this process, each query's code is manually written according to the sequential flow of database operators. That is, each query undergoes a manual pass phase to generate C++ code, ensuring that the output results are consistent with the results produced by PostgreSQL.

Data Reading and Storage. In our designed architecture, we use the importData function within the custom Table class to achieve data reading from disk to memory. The importData function accepts two parameters: one is the structure of the table, which records the data types of each column, and the other is the data file address (ending in .tbl). This method splits the content of the given file line by line according to the structure of the table and places it into a vector container. If additional information is needed, such as column names or table names, they can be obtained by calling the methods within the Table class. In future work, we will use Arrow for storage and reading, which will make the process more efficient (Fig. 2).

Operators. We implement essential SQL operators such as Scan, Filter, Join, GroupBy, and Sort within our framework using the C++ standard library.

**Fig. 2.** importData**Fig. 3.** queryCompile

Specifically, Scan operations are realized by directly accessing data stored in the Table class's vector. Filter operations utilize standard C++ iteration and lambda functions for efficient traversal and condition application. For Join operations, we provide multiple implementations, including hash joins utilizing unordered_map for efficient data matching, alongside nested loop and sort-merge join algorithms for flexibility. GroupBy operations utilize std::map to create a map where the key is the column being grouped, and the value is an aggregation of the corresponding values. Based on this map, different aggregation functions (e.g., MIN, MAX, STDEV) can be easily applied. Finally, Sort operations leverage std::sort for customizable ordering. All data structures used in our implementation are based on the C++ standard library, ensuring compatibility and ease of use.

Query Execution Plan. Our implementation also adopts the process used in psql when generating execution plans with EXPLAIN ANALYZE. The sequence follows the general order of data filtering, join operations, grouping by certain attributes, and then ordering by specified criteria.

Pre-Compilation of Dependencies. We employ two distinct compilation methods in our framework: AOT compilation and LLVM JIT compilation. In AOT compilation, we directly compile the generated C++ code into an executable file using clang++. This file is then executed to process the TPCH data and produce the query results. In LLVM JIT compilation, we take a different approach. First, we compile the C++ code into LLVM bitcode, an intermediate representation. Then, we utilize the LLVM lli tool, which leverages JIT compilation to dynamically optimize and run the bitcode, processing the TPC-H data and generating the results.

For both methods, we meticulously record the compilation time and the execution time for performance comparison. As shown in Fig. 3, to optimize the compilation process, we precompile the header files (Table.h and Structs.h) and then link them with the compiled query code.

4 Experiments

Environment. Our experiments were conducted on an Intel i7 processor with 16 GB of RAM, running Ubuntu 20.04.1. We utilized Clang and LLVM's JIT

compiler both with optimization level -O3 enabled for code compilation. We compare our implementation with PostgreSQL 12.17, configured for serial execution to ensure a fair comparison with the handwritten C++ code, which did not employ parallel libraries.

Benchmark Comparison. For threads, since our CPP code does not involve parallel computation and processing, we need to disable the default parallel processing in PSQL to ensure that time measurements are based on single-threaded processing. For execution time, we use the EXPLAIN ANALYZE command to record the SQL execution time in PSQL, excluding its parsing time, to compare the differences in their data processing capabilities.

Dataset. We used three different dataset sizes: 0.01 GB, 0.1 GB, and 1 GB. These datasets, generated using the TPC-H tool, include tables such as customer, supplier, orders, lineitem, part, employee, nation, region, and transaction. To ensure a fair comparison with our C++ implementation, we utilized a properly tuned PostgreSQL database with appropriate indexes built on relevant columns.

Queries. We selected 22 TPC-H queries, which are business-oriented and designed to evaluate the performance of database systems. When making comparisons, we selected queries that are particularly distinctive as examples. For instance, query1 is a single-table statistical query, query3 is a multi-table statistical query, query4 is a nested query, and query 13 involve GROUP BY and ORDER BY clauses.

Metrics. We use the TPC-H benchmark to evaluate the performance differences between handwritten C++ code and traditional database systems. The TPC-H benchmark is a standard tool for assessing database performance, comprising a set of business-oriented ad-hoc queries and concurrent data modifications. TPC-H queries are designed to stress-test databases in terms of throughput and response time, making it a valuable tool for performance evaluation.

Exp 1. Execution Time. First, we compared the execution times of AOT and JIT with the execution time of running queries in PSQL under different data volumes. As can be seen in Fig. 4, when comparing execution times for different data volumes, the execution times of AOT and JIT are much shorter than the execution time of PSQL. Without considering the inclusion of compilation time, JIT has the shortest execution time. Among these, the average execution time of JIT is approximately 90% of AOT and 20% of the execution time of PSQL in Fig. 4.

Exp 2 - Compilation Time. In this experiment, we compared the compile time and the time to generate bitcode. The result is demonstrated in Fig. 4 (d). In the compilation figure, we can see that the time spent by JIT is shorter than the compilation time of AOT (both under optimization level 3). The compile time of JIT is only about 66% of that of AOT.

Exp 3 - Varying Data Volumes. It can be seen in Fig. 5 that as the data volume increases, the time taken by JIT becomes increasingly shorter compared

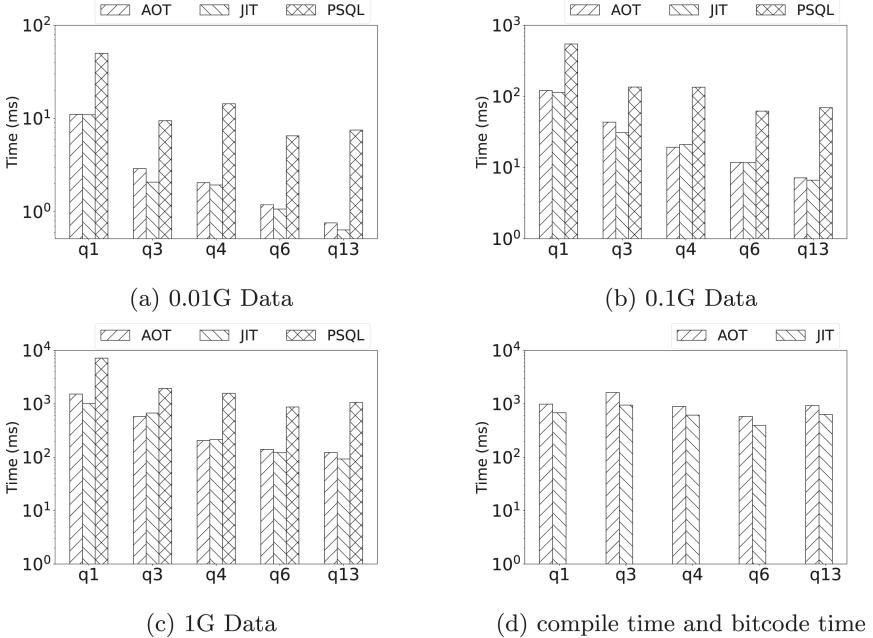


Fig. 4. Execution Time Comparison and Compilation Time Comparison

to PSQL. We selected query1, a single-table statistical query, as a typical case to estimate the speed of JIT versus PSQL under different data volumes through linear regression. Query3 is a multi-table statistical query, and the results are similar. As shown in Fig. 5, we found that when the data volume is small, the compilation time makes JIT not have a significant advantage. However, as the data volume exceeds 0.1G, the compilation and execution speed of JIT becomes faster than that of PSQL, with their slope ratio reaching up to seven times.

5 Future Work

Automatic SQL to Code Translation: Develop mechanisms for automatically translating SQL queries into executable code. This will involve creating a parser and translator that can handle various SQL syntax and generate efficient, optimized code.

Applying MLIR Technology to the Compilation Pipeline: This is a relatively novel technology that addresses the issue of compilation span. It involves repeatedly refactoring and optimizing our SQL code, ultimately transforming it into LLVM IR and performing JIT compilation.

Expansion of SQL Operators: Extend the range of supported SQL operators to enhance the system's versatility and applicability to a wider range of

query types. This includes implementing advanced SQL features such as window functions, complex joins, and subqueries.

Parallel and Vectorized Execution: Investigate techniques for parallel and vectorized execution to leverage modern hardware capabilities of and further boost query performance. This involves designing algorithms that can efficiently distribute workload across multiple cores and utilize SIMD (Single Instruction/Multiple Data) instructions.

Integration with Databases: Explore methods for integration of the JIT compilation framework with existing database systems, enabling enhanced performance and functionality. This will require developing interfaces and protocols to ensure compatibility with popular database engines like PostgreSQL, and others.

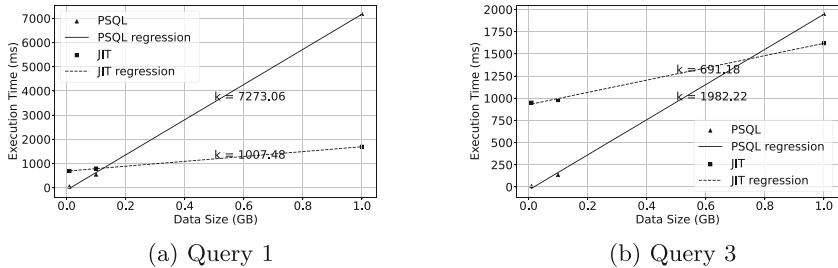


Fig. 5. Comparison of PSQL and JIT Time Regression Analysis

JIT-Tailored Optimizer: Design and implement an optimizer specifically tailored for JIT-compiled execution, including the redesign of cost estimation functions. This optimizer will leverage the dynamic nature of JIT compilation to produce better query execution plans on the current workload and hardware.

JIT Compilation Cache: Develop a caching mechanism for JIT-compiled code to reuse previously generated machine code, reducing compilation overhead and improving query execution times. This will involve creating a caching strategy that can efficiently store and retrieve compiled queries and code segments.

6 Conclusion

In this paper, we presented our preliminary work on a proof-of-concept system leveraging LLVM's JIT compilation capabilities to enhance database query execution. By implementing SQL queries in C++ and compiling them into optimized machine code, we demonstrated the feasibility and potential of a simple, lightweight architecture for JIT-based database systems. Our approach reduces engineering effort, improves observability, and enhances system stability compared to more complex implementations like HyPer.

Acknowledgments. Zhengyi Yang is supported by Enmotech Data AU.

References

1. LLVM. <https://www.llvm.org/>
2. TPC-H benchmark. <https://www.tpc.org/tpch/>
3. Akeret, J., Gamper, L., Amara, A., Refregier, A.: Hope: a python just-in-time compiler for astrophysical computations. *Astron. Comput.* **10**, 1–8 (2015)
4. DeBrabant, J., Pavlo, A., Tu, S., Stonebraker, M., Zdonik, S.: Anti-caching: a new approach to database management system architecture. *Proc. VLDB Endow.* **6**, 1942–1953 (2013)
5. Graefe, G.: Volcano - an extensible and parallel query evaluation system. *IEEE Trans. Knowl. Data Eng.* **6**, 120–135 (1994)
6. Haffner, I., Dittrich, J.: A simplified architecture for fast, adaptive compilation and execution of SQL queries. In: *EDBT*, pp. 1–13 (2023)
7. Imasheva, B., Azamat, N., Sidelkovskiy, A., Sidelkovskaya, A.: The practice of moving to big data on the case of the NoSQL database, clickhouse. In: Le Thi, H.A., Le, H.M., Pham Dinh, T. (eds.) *WCGO 2019. AISC*, vol. 991, pp. 820–828. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-21803-4_82
8. Kemper, A., Neumann, T.: Hyper: Hybrid oltp and olap high performance database system. Technical report (2010)
9. Kohn, A., Leis, V., Neumann, T.: Adaptive execution of compiled queries. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 197–208. IEEE (2018)
10. Leis, V., Boncz, P., Kemper, A., Neumann, T.: Morsel-driven parallelism: a NUMA-aware query evaluation framework for the many-core age. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 743–754 (2014)
11. Ma, M., Yang, Z., Hao, K., Chen, L., Wang, C., Jin, Y.: An empirical analysis of just-in-time compilation in modern databases. In: *Databases Theory and Applications*, pp. 227–240 (2024)
12. Melnik, D., Buchatskiy, R., Zhuykov, R., Sharygin, E.: JIT-compiling SQL queries in PostgreSQL using LLVM. In: *Proceedings of the PostgreSQL Conference* (2017)
13. Neumann, T.: Efficiently compiling efficient query plans for modern hardware. *Proc. VLDB Endow.* **4**(9), 539–550 (2011)
14. Neumann, T., Freitag, M.J.: Umbra: a disk-based system with in-memory performance. In: *CIDR*, vol. 20, p. 29 (2020)
15. Neumann, T., Freitag, M.J.: Umbra: a disk-based system with in-memory performance. In: *Conference on Innovative Data Systems Research* (2020)
16. Rubinsteyn, A., Hielscher, E., Weinman, N., Shasha, D.: Parakeet: a {Just-In-Time} parallel accelerator for python. In: *4th USENIX Workshop on Hot Topics in Parallelism (HotPar 12)* (2012)
17. Suganuma, T., et al.: Overview of the IBM java just-in-time compiler. *IBM Syst. J.* **39**(1), 175–193 (2000)
18. Wade, A., Kulkarni, P., Jantz, M.: AOT vs. JIT: impact of profile data on code quality. *ACM SIGPLAN Not.* **52**, 1–10 (2017)

Author Index

A

Ai, Chunyu 210

B

Bao, Xuguang 393, 398

C

Cao, Yu 17

Che, Haoyang 481

Chen, Deng 3

Chen, Fangshu 383

Chen, Lina 165

Chen, Liuyi 357

Chen, Rui 113

Chen, Xiaoshuang 504

Chen, Xiye 95

Chen, Yilu 17

Chen, Zhenghua 448

Chen, Zhiwei 393

Cheng, Yide 80

Chzhen, Di Yuan 408

Cui, Shuangshuang 368

Cui, Zhe 113

D

Ding, Linlin 408

Ding, Shuaipeng 130

Ding, Xuanang 46

Ding, Yi 357

Dong, Guozhong 378

Dong, Zhaoan 403

Dong, Zhenjiang 415

F

Fan, Dongyi 3

Feng, Yuechun 240

Florescu, Corina 63

G

Gao, Hong 165

Ge, Ningchao 347

Geng, Zezheng 393

Gu, Zhaoquan 17

Guo, Hao 378

Guo, Hua 459

Guo, Jiangpu 438

Guo, Longjiang 210, 255

H

Han, Bocheng 504

Han, Yinjun 448

Hao, Jiaqi 210

Hao, Kongzhang 357, 504

Hao, Yu 504

He, Lili 3

He, Xiaohua 286

Hu, Ning 17

Huang, He 302

Huang, Hongbin 347

J

Ji, Yunhong 459

Jia, Yan 17

Jiang, Kun 224

Jiang, Qijun 165

Jiao, Qingju 403

Jin, Hu 330

Jin, Wei 63

Jin, Yi 504

K

Kong, Xiangjie 286

L

Li, Bohan 492

Li, Chenguang 368

Li, Guangshun 403

Li, Guohui 46

Li, Jinxuan 368
 Li, Mingyong 130, 146
 Li, Mo 408
 Li, Peng 255
 Li, Qinyuan 481
 Li, Shunyang 357
 Li, Xuan 347
 Li, Yang 80
 Li, Yanzeng 362
 Li, Yuda 408
 Li, Zhenyu 179
 Lin, Dekun 113
 Lin, Hang 286
 Lin, Yuting 388
 Liu, Boge 504
 Liu, Dingwei 179
 Liu, Lihua 347
 Liu, Yi 3
 Liu, Yongfei 357
 Liu, Zheyng 398
 Lu, Xiaoyu 438

M

Ma, Miao 504
 Ma, Yan 130
 Mei, Yihan 31
 Mu, Lin 80

N

Nguyen, Vanluan 352
 Ning, Haoran 357, 504

P

Pan, Haiwei 240, 318
 Peng, TaiLai 113

Q

Qi, Tianlong 210

R

Ren, Bingqing 195
 Ren, Jingbiao 368
 Ren, Meirui 210, 255
 Rong, Qian 46

S

Shang, Biyun 415
 Shao, Bingdi 255
 Shen, Guojiang 286

Shi, Haobin 427
 Shi, Ji 448
 Shil, Avijeet 63
 Shu, Xinsheng 146
 Shui, Jianan 130
 Sun, Hailong 438
 Sun, Jun 271
 Sun, Qindong 224
 Sun, Xinyuan 448

T

Tian, Yixin 383
 Tong, Yiqiu 165
 Tran, Vanha 352
 Tu, Chenfeng 438

W

Wang, Benfeng 286
 Wang, Bingkun 383
 Wang, Binyu 302
 Wang, Chunling 504
 Wang, Dong 368
 Wang, Hai 470
 Wang, Hongzhi 368
 Wang, Jiahui 383
 Wang, Liping 302
 Wang, Lizhen 373
 Wang, Peng 95
 Wang, Shaolin 481
 Wang, Shuai 470
 Wang, Tengyun 347
 Wang, Xiaofan 403
 Wang, Xiaoling 31
 Wang, Xiaoyu 80
 Wang, Xieyang 388
 Wang, Xinyu 31
 Wang, Ye 17

Wei, Keai 255
 Wu, Hao 368
 Wu, Hualong 378
 Wu, Jibing 347
 Wu, Weijie 165
 Wu, Wenlong 492

X

Xie, Xinran 113
 Xie, Zhiran 408
 Xiong, Jing 403
 Xu, Jianqiu 388

- Xu, Junning 415
Xu, Mo 415
Xu, Zekun 492
- Y**
Yang, Dongmei 195
Yang, Hongzhang 438
Yang, Longze 330
Yang, Peizhong 373
Yang, Peng 195
Yang, Yan 330
Yang, Zhengyi 357, 504
Yang, Zhengyu 398
Yao, Lei 481
Yi, Meng 195
Yi, Weipo 210
Yi, Yan 481
Yin, Zhanzuo 492
Yuan, Ling 46
- Z**
Zhan, Yirui 362
Zhang, Chengwei 504
Zhang, Dell 31
Zhang, Feng 415
Zhang, Kejia 240, 318
Zhang, Lichen 255
Zhang, Lingli 373
Zhang, Lu 46
Zhang, Minhao 362
Zhang, Suqiong 3
Zhang, Tianming 240
Zhang, Wanchuan 504
Zhang, Yiping 427
Zhang, Yitong 388
Zhang, Yiwen 80
Zhang, Yuang 302
Zhang, Yufei 383
Zhang, Zhibin 179
Zhang, Zhiyong 408
Zhao, Hui 271
Zhao, Wei 368
Zhao, Xinzhe 492
Zhou, Lihua 373
Zhou, Xiaolei 470
Zhou, Xuan 459
Zhu, Lei 224
Zhu, Shaoqiang 318
Zhuo, Junnan 492
Zou, Lei 362